LEARNING VISUAL TASKS WITH SELECTIVE ATTENTION

BY

KEVIN JONATHAN SHIH

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Doctoral Committee:

Associate Professor Derek Hoiem, Chair
Associate Professor Svetlana Lazebnik
Professor David Forsyth
Assistant Professor Devi Parikh, Georgia Tech

ABSTRACT

Knowing where to look in an image can significantly improve performance in computer vision tasks by eliminating irrelevant information from the rest of the input image, and by breaking down complex scenes into simpler and more familiar sub-components. We show that a framework for identifying multiple task-relevant regions can be learned in current state-of-the-art deep network architectures, resulting in significant gains in several visual prediction tasks. We will demonstrate both directly and indirectly supervised models for selecting image regions and show how they can improve performance over baselines by means of focusing on the right areas.

ACKNOWLEDGMENTS

I would like to start by thanking my advisor, Derek Hoiem, as I wouldn't have made it so far without his guidance. Throughout my time here, he has provided remarkably keen insight to the problems I was working on, and ultimately taught me how to be a better researcher. I would also like to thank other members of the computer vision faculty here: David Forsyth, Svetlana Lazebnik, and more recently Alexander Schwing for the numerous useful and informative discussions I've had with them throughout the years.

Next, I would like to thank all of my labmates, past and present, for creating a great environment for improving myself as a grad student. Thanks to Ian Endres for showing me the ropes when I was getting started. Daphne Tsatsoulis always offered great words of encouragement (and baked goods) when things weren't quite working out. Arun Mallya and Saurabh Singh both helped significantly in getting my feet wet with deep learning, and they also contributed significantly to the methods described in this thesis. I would also like to thank Tanmay Gupta for our collaboration – the last chapter would not have been possible without him, Liwei Wang for the many late-night chats we've had in our shared office, Bryan Plummer for always providing great feedback for research ideas and for lecturing me about the world of hip hop, Aditya Deshpande and Arun (again) for taking over the server management duties from me, and honorary vision lab member Yonatan Bisk for offering sage advice and for sending me a bottle of ghost pepper vodka that I have yet to figure out how to safely consume.

Finally, I would like to thank my friends and family. Special thanks go to Jonathan Ligo for letting me pick his brain on various topics ranging from academic to culinary, Pooya Khorrami for always being supportive and for the delicious bucket of pineapple cotton candy, Andrew Murphy for being a great roommate for the last few years, and Chris Cervantes for sharing my interest in games despite the lack of time to play them. I would especially like

to thank Nadia Danienta for supporting me through everything over the last two years. Most importantly, I would like to thank my family for supporting me through my incredibly long schooling process. I'm not entirely sure how they managed to financially support my undoubtedly expensive education, but I'm very thankful for the sacrifices they made to make it happen.

TABLE OF CONTENTS

LIST OF TABLES vii

LIST OF FIGURES ix

LIST OF ABBREVIATIONS xv

CHAPTER 1 INTRODUCTION 1
1.1 Contributions 3
1.2 Original Publications 4

CHAPTER 2 BACKGROUND 6
2.1 Visual Attention and Saliency 6
2.2 Vision Tasks 12

CHAPTER 3 LOCALIZING KEYPOINTS 16
3.1 Method 17
3.2 Results 21
3.3 Conclusion 23

CHAPTER 4 FINE GRAINED CLASSIFICATION WITH ALIGNED PARTS 24
4.1 From Keypoints to Regions 24
4.2 Fine-Grained Classification 26
4.3 Conclusion 27

CHAPTER 5 LATENT ATTENTION FOR VISUAL QUESTION ANSWERING 29
5.1 Introduction 29
5.2 Approach 31
5.3 Experiments 36
5.4 Conclusion 44

CHAPTER 6 VISUAL ATTENTION FOR VISUAL QUESTION ANSWERING WITH SUPERVISED PHRASE GROUNDING . . . 45
6.1 Method 46
6.2 Experiments 54
6.3 Conclusion 60

CHAPTER 7 CONCLUSIONS 63

CHAPTER 8 REFERENCES 65

LIST OF TABLES

3.1 Localization and Visibility Prediction Performance of various methods without using the ground truth Bounding Box 22

3.2 Comparison of per-part PCP with Liu *et al* 2013 [1] and Liu *et al* 2014 [2]. The abbreviated part names from left to right stand for back, beak, belly, breast, crown, forehead, eye, leg, wing, nape, tail, and throat. 23

4.1 Comparison of Part Localization Performance: Our method based on keypoint prediction from Edge Boxes shows significant improvement over previous work. 25

4.2 Comparison of our classification with other works 27

5.1 Overall accuracy comparison on Validation. Our region selection model outperforms our own baselines, demonstrating the benefits of selective region weighting. 37

5.2 Accuracy comparison on VQA test sets. 40

5.3 Accuracies by type of question on the validation set. Percent accuracy is shown for each subset for our region-based approach, classification using only text, text with a whole-image feature vector, and text with salient attention (attention based only on image). Overall, our region selection scheme outperforms use of whole images by 2% and text-only features by 5%. The learned salient attention model performed surprisingly well. Most notably, it had a similar performance boost over the whole-image baseline on scene questions. The proposed region-selection model still outperforms all baselines on color questions, suggesting that attention for color-identification cannot be easily learned via saliency. 41

5.4 Language model comparison. The 2-bin model is the concatenation of the question and answer averages. The parsed model uses the Stanford dependency parser to further split the question into 4 bins. 42

6.1 **Inductive transfer from VR to VQA through SVLR in joint training:** We evaluate the performance of our model with the SVLR module trained jointly with VR and VQA supervision (provided by Genome and VQA datasets respectively) on the validation set of the multiple-choice VQA task. We compare this jointly-trained model to a model trained on *only* VQA data. We also compare to a traditional multitask learning setup that is jointly trained on VQA and VR and shares visual features but *does not* use the object and attribute word embeddings for recognition. While multitask learning outperforms the VQA-only model, using the SVLR module doubles the improvement. Our model is most suited for the question types in bold that require visual recognition without specialized skills like counting or reading. In this setting we train on Genome VR data and apply to VQA val. Details in Sec 6.2.2. 54

6.2 **VQA performance on val and test sets:** Because these systems vary in many ways, our internal comparisons are more instructive, but we include these for reference. For test accuracy, it is unclear whether FDA uses *val* in training. The MLP results were obtained using the implementation provided by [3]. The original MLP implementation [4] using Resnet-101 yields 64.9 and 65.2 on *test-dev* and *test-std* respectively. MCB reports only *test-dev* accuracy for the directly comparable model (final without ensemble). . . 59

LIST OF FIGURES

2.1 Top salient Edge Boxes [5] on an example image. Region proposals allow us to avoid unecessarily process uninteresting background regions 8

3.1 The pipeline of our keypoint localization process: Given an input image, we extract multiple edge boxes. Using each edge box, we make predictions for the location of each of the 15 keypoints, along with their visibility confidences. We then find the best predicted location by performing confidence thresholding and finding the medoid. The process is illustrated for the right eye keypoint (Black edge boxes without associated dots make predictions with confidences below the set threshold, and green is an outlier with a high confidence score). 19

3.2 Qualitative results for a subset of the keypoints. Predictions for most of the images cluster tightly. Therefore, simple prediction methods such as medoids work well. Medoid shift adds to the robustness, leading to further improvements (second last column). Primary failure mode is when visibility thresholding fails to rule out clusters of false positives (bottom right). 22

4.1 Examples of good (left) and failed (right) localization results: The ground truth boxes are in solid black. The head, torso, and whole body boxes are in green, blue and red respectively. The head is correctly localized in most of the above examples. In the top row middle example, even though the whole body box IOU is low, most of the missed area is actually background due to the bird extending its wings. In the bad examples, we show that we mostly fail in rare close-ups and when there are multiple instances. 26

4.2 Camporison of classification accuracies obtained using varying combinations of parts localized under different conditions . 27

5.1 Our goal is to identify the correct answer for a natural language question, such as "What color is the walk light?" or "Is it raining?" We focus on the problem of learning where to look. The above figure shows example attention regions produced by our model. 30
5.2 Examples from VQA ([6]). From left to right, the above examples require focused region information to pinpoint the dots, whole image information to determine the weather, and abstract knowledge regarding relationships between children and stuffed animals. 30
5.3 Overview of our network for the example question-answer pairing: "What color is the fire hydrant? Yellow." Question and answer representations are concatenated, fed through the network, then combined with selectively weighted image region features to produce a score. 31
5.4 Example parse-based binning of questions. Each bin is represented with the average of the word2vec vectors of its members. Empty bins are represented with a zero-vector. . . . 35
5.5 Comparison of salient attention, conditioned on only the image, and the proposed attention model that considers both image and query. Many images have predictable saliency, in that it is easy to predict what any question in the image will be about. In the top row of this figure, the salient object is the plane and is predicted by both models with and without considering the query text. In more complex cases such as the bottom two rows, where there are multiple foreground objects, the salient model does a decent job of identifying those over the background, but fails to produce the correct attention map when the query refers to only one of the many possible foreground objects. 39
5.6 Comparison of attention regions generated by various question-answer pairings for the same question. Each visualization is labeled with its corresponding answer choice and returned confidence. We show the highlighted regions for the top multiple choice answers and some unrelated ones. Notice that in the first example, while the model clearly identified a green region within the image to match the "green" option, the corresponding confidence was significantly lower than that of the correct options, showing that the model does more than just match answer choices with image regions. 40
5.7 Example image with corresponding region weighting. Red boxes correspond to manual annotation of regions relevant to the question: "Are the people real?" 42

5.8 Comparison of qualitative results from Val. The larger image shows the selection weights overlaid on the original image (smaller). L: Word only model; I: Word+Whole Image; R: Region Selection. The scores shown are ground truth confidence - top incorrect. Note that the first row shows successful examples in which tight region localization allowed for an accurate color detection. In the third row, we show examples of how weighting varies on the same image due to differing language components. 43
5.9 Plot of color-based question accuracy with varying number of regions sampled at every 10. The experiment was run on a 10% held-out set on train. We look at using the weighted average of only the top K scoring regions, as well as only the Kth. We include the whole image baseline's accuracy in this category for comparison. 44

6.1 **Sharing image-region and word representations across multiple vision-language domains:** The SVLR module projects images and words into a shared representation space. The resulting visual and textual embeddings are then used for tasks like Visual Recognition and VQA. The models for individual tasks are formulated in terms of inner products of region and word representations enforcing an alignment between them in the shared space. 46
6.2 **Joint Training on Visual Recognition(VR) and Visual Question Answering(VQA) with the proposed SVLR Module:** The figure depicts sharing of image and word representations through the SVLR module during joint training on object recognition, attribute recognition, and VQA. The recognition tasks use object and attribute labelled regions from Visual Genome while VQA uses images annotated with questions and answers from the VQA dataset. The benefit of joint training is that while the VQA dataset does not provide region groundings of nouns and adjectives in the QA (eg. "fluffy","dog"), this complementary supervision is provided by the Genome recognition dataset. Models for each task involve image and word embeddings produced by SVLR module or their inner products (See Fig 6.3 for VQA model architecture). 47

6.3 **Inference in our VQA model:** The image is first broken down into Edge Box region proposals[5]. Each region R is represented by visual category scores $s(R) = [s_o(R), s_a(R)]$ obtained using the visual recognition model. Using the SVLR module, the regions are also assigned an attention score using the inner products of region features with representations of nouns and adjectives in the question and answer. The region features are then pooled using the relevance scores as weights to construct the *attended* image representation. Finally, the image and question/answer representations are combined and passed through a neural network to produce a score for the input question-image-answer triplet. 50

6.4 **Interpretable inference in VQA:** Our model produces interpretable intermediate computation for region relevance and object/attribute predictions for the most relevant regions. Our region relevance explicitly grounds nouns and adjectives from the Q/A input in the image. In addition to attention, we show object and attribute predictions for the most relevant region identified for a few correctly answered questions. The relevant regions are visualized by applying a mask generated from relevance scores projected back to their source pixel locations. 53

6.5 **Inductive Transfer from VQA to Object Recognition:** Each cell's color reflects the average accuracy improvement for classes within the corresponding frequency ranges of both datasets from training on Genome-only to training on Genome and VQA. Most gains are in rare Genome nouns with higher frequency in the VQA dataset (top left corner), suggesting that the weak supervision provided by training VQA attention helped to augment performance via the SVLR. The numbers in each cell show the Genome-only mean accuracy +/- the change due to SVLR multitask training, followed by the number of classes in the cell in parentheses. 57

6.6 **Failure modes:** Our model cannot count or read, though it will still identify the relevant regions. It is blind to relations and thus fails to recognize that *birds*, while present in the image, are not *drinking water*. The model may give a low score to the correct answer despite accurate visual recognition. For instance, the model observes *asphalt* but predicts *concrete*, likely due to language bias. A clear example of an error due to language bias is in the top-left image as it believes the lady is holding a *baby* rather than a *dog*, even though visual recognition confirms evidence for dog. Finally, our model fails to answer questions that require complex reasoning involving comparison of multiple regions. 58

6.7 Synthetic center-focused image baseline provided by the authors of Das *et al* [7]. This image was used to represent a baseline attention model that always focuses on the center of the image. By computing the correlation between the human attention maps and this one, we are able to identify low correlation subsets of the dataset in which the human subjects looked away from the image center. 60

6.8 Qualitative comparison of attention maps from various models. Saliency generally corresponds pretty well with what questions ask about. Compared to the WTL model, the SVLR model's attention is typically much more focused. Regions deemed irrelevant by the SVLR seem to be more readily downweighted than in the WTL and Salient cases. Note that Gaussian smoothing was used on the attention masks for Salient, WTL, and SVLR for visualization purposes only. 61

6.9 Mean Spearman rank-correlation coefficients between model attention and human attention at various threshold. The threshold points define subsets of the dataset for which the human attention correlation with the synthetic center heatmap is below the current threshold value. For example: the first sample point of each curve is the mean correlation of each model with human attention, measured on a subset in which the human attention's correlation with the center heatmap is less than or equal to 0. WTL and Salient are the proposed model and salient attention baseline from the previous chapter. The Center baseline is the correlation of the center heatmap measured against all examples in the current subset. As can be seen, the attention of the proposed SVLR model significantly outperforms those of the models from the previous chapter at all threshold levels. The WTL slightly outperforms its corresponding strong salient baseline up to the threshold at 0.6. As the threshold approaches 1, the synthetic center heatmap baseline outperforms all proposed models, confirming that the majority of the questions are asking about something in the center of the image. Note that there were only 11 examples in ≤ 0 threshold and 748 in the ≤ 0.6 threshold. 62

LIST OF ABBREVIATIONS

VQA	Visual Question Answering
VR	Visual Recognition
HOG	Histogram of Oriented Gradients
SVM	Support Vector Machines
CNN	Convolutional Neural Networks
LSTM	Long Short-Term Memory
LDA	Linear Discriminant Analysis

CHAPTER 1

INTRODUCTION

Consider what happens when people attempt to recognize a face. Do they observe every visible component of the face with the same amount of attention? Or do they spend more time looking for distinctive features such as a mole on the cheek, the shape of the eyes, or even the contour of their jawline? Visual attention is focusing the visual system on what is most informative and relevant for the task at hand. The following work addresses the problem of training computer vision models capable of exploiting visual attention to improve their own performance.

Visual attention in computer vision systems is strongly motivated by the way the human visual system works. In 1967, Alfred Yarbus [8] was able to track the gaze of his human subjects as they observed certain images, noting that most of the attention was directed towards parts of the image that the subject considered to be most informative. While computer vision systems are designed for performance rather than to simulate their analogs in biology, the main takeaway is that not all of the visual input is equally important. Reasons to adopt attention-like behavior can be for computational reasons (much of the input image can be ignored) or for improving accuracy (irrelevant parts of the image may distract the model).

To date, incorporating visual attention has produced many successful models in the field of computer vision. One such example is fine-grained image recognition. The human attention analog for fine-grained recognition can be observed when one tries to identify the difference between two similar objects. Consider the act of comparing two different species of magpies. In order to compare, the human gaze will likely dart back and forth between analogous parts on both birds, comparing the eyes, beaks, and breast pattern, wings etc. to pick up minute differences. The state of the art in fine-grained image recognition has taken a similar approach to comparing similar objects. When comparing birds, analogous regions such as the head and torso are first

localized in all instances, then part-specific classifiers are trained to classify the birds based on just the head or the bird. In this setting, the specialized classifiers are able to learn the minute details that distinguish the head of one species of bird from another's – something that would have otherwise been difficult to learn from whole-images of birds in various poses and orientations.

Another successful application of attention is in the recognition of complex objects and scenes. Consider the task of recognizing a restroom. On the one hand, one could go through the difficult process of attempting to model the full visible appearance of a restroom, including the 3D geometry and all possible positionings of the sink, bathtub, and toilet, as well as the color of the tiles and walls. Alternatively, one could note that some of the above attributes are more important than others – knowing the color of the wall says little about whether you are looking at a restroom, but the presence of a toilet alone is a strong and often sufficient indicator. Additionally, identifying the presence of a toilet in a bathroom or a bookshelf in a bookstore requires only a detector, significantly simplifying the approach. As such, modeling recognition as the detection of a few highly discriminative visual patterns has seen widespread adoption. It has also spawned an interesting line of work dealing with the discovery of these dicriminative parts and patches.

An interesting complication to consider is what if we were dealing with a collection of different tasks, each requiring different behavior for visual attention? This is an important problem to consider, as the push to develop more human-like AIs will necessitate having a model that can adapt to new tasks and situations. The latter half of this work specifically addresses the problem of visual attention for visual question answering, in which questions about images may pose a variety of different tasks. In this setting, it is up to the model to adjust its attention behavior to best handle the currently posed question.

While our computer vision models will ultimately process whatever we show them, being selective about where we make them look can be advantageous in many scenarios. The goal of this work is to demonstrate several ways of incorporating selective visual attention into models for various computer vision tasks.

1.1 Contributions

We propose trainable models capable of identifying image regions relevant to their respective tasks. Our models are applied to part localization, fine-grained image recognition, and visual question answering (VQA), the last of which can be considered to be a meta-vision task. With identifying task-relevant image regions as the broader picture, the problems we tackle in the following works are as follows:

Localizing Parts and Keypoints with Multiple Crops: Part and keypoint localization can be conditioned on the nearby context within the image. For example, the necessary information to localize an eye would lie on the face. In order for a localization model to accurately predict the location down, we would ideally like to feed the model an input image with as much resolution as possible. However, due to some CNN-based architectures having a fixed input resolution (e.g. 224x224 pixels), we would ideally feed in the minimum necessary context into the model at as high a resolution as possible, as any unecessary context would come at the cost of reduced resolution for the necessory context. In chapter 3, we introduce a sampling-based approach to identifying the best image regions from which to make localization predictions. By conducting the prediction task on multiple random crops from the image, we expect at least some of the crops to be close to the optimal context region. We then propose a simple scoring scheme that, combined with outlier rejection, allows us to identify a robust set of candidate predictions from which to predict the final keypoint location. Further, our candidate identification also allows us to accurately predict when a keypoint *is not* present in the image at all.

Part-aligned Fine-Grained Classification: As demonstrated in previous works, aligning analogous regions across images is an effective strategy for fine-grained classification. We demonstrate that a keypoint-localization method that can accurately predict both position and visibility leads to very accurate alignments and, by extension, better classification.

Conditioning Visual Attention on Language: Building machines that can interface with natural language instructions is an end-goal of human-robot interaction. Leveraging advancements in deep learning frameworks, we are interested in directing the visual attention of a vision system with natural language. Specifically, given the natural queries such as "What color is the

car?" or "Is there a cat on the bed?", the system should focus on question-relevant regions of the image to answer the queries. The main complication is that question-relevant region annotation may not exist. As such, we are interested in training such a model using only question-answer supervision. By incorporating question-relevant region selection as a latent task, we expect the model to learn region-question relevance as a means of improving its question answering accuracy.

Language-based Visual Attention with Phrase-Level Supervision: While question-level attention is hard to supervise, it is certainly feasible to provide supervision for individual phrases within the question. Phrase-level attention cannot always tell you exactly where to look, specifically if the target object is never directly mentioned. For example, to answer "Is something sitting on the chair?", we cannot train a model to localize "something" in the image directly. However, we can first localize the mentioned "chair" and use that to aid the search process. Further, phrase-level attention can be trained using existing datasets for object detection and phrase-localization, allowing us to introduce additional supervision into the visual question answering task at a lower level.

1.2 Original Publications

The chapters in this work are based following original publications and tech reports:

- *Shih, Kevin J., Arun Mallya, Saurabh Singh, and Derek Hoiem. "Part localization using multi-proposal consensus for fine-grained categorization." BMVC 2015.* (Chapters 3 and 4) As primary author, my contributions include the design and experimentation of the outlier removal and consensus method from multiple predictions. Co-author Arun Mallya contributed significantly to the implementation of the CNN whereas co-author Saurabh Singh proposed the use of medoid-shift over simple medoids.

- *Shih, Kevin J., Saurabh Singh, and Derek Hoiem. "Where to look: Focus regions for visual question answering." CVPR 2016.* (Chapter 5) As primary author of this work, my contributions include the majority

of the implementation for both the model and experimentation. The formulation of the attention model was jointly derived with co-authors.

- *Gupta, Tanmay, Kevin Shih, Saurabh Singh, and Derek Hoiem. "Aligned Image-Word Representations Improve Inductive Transfer Across Vision-Language Tasks." arXiv preprint arXiv:1704.00260 (2017).* (Chapter 6) Tanmay Gupta is the primary author of this work. My specific contributions involve design decisions regarding the loss functions, the design of the word to region embedding mechanism, inductive transfer analysis from VQA to object recognition, and the additional human attention comparison.

CHAPTER 2

BACKGROUND

This chapter provides background for the technical concepts in this work. We begin with an overview of visual attention in existing literature, followed by background for the specific tasks addressed in our work.

2.1 Visual Attention and Saliency

Our work focuses on applying the concept of visual attention and saliency to various computer vision tasks. In brief, visual attention refers to selectively attending to relevant parts of the input, and saliency is the extent to which something in the input stands out or will be attended to.

Visual attention has long been an important topic of study in human cognition. In 1967, Yarbus [8] studied how people's eyes moved when perceiving complex objects by attaching measurement devices to the eye. He noted that "When looking at a human face, an observer usually pays most attention to the eyes, the lips, and the nose. The other parts of face are given much more cursory consideration." In other words, the human visual system will specifically focus on the most salient and informative parts of the visual input.

Later works attempted to model the human attention system. Two of the most influential works in this field are the Feature Integration Theory (FIT) of Treisman and Gelade [9] and the Guided Search model of Wolfe *et al* [10, 11]. The FIT suggests a bottom-up pipeline such that: "features are registered early, automatically, and in parallel across the visual field, while objects are identified separately and only at a later stage, which requires focused attention." The Guided Search model was later proposed to address some issues with the FIT, specifically that top-down information can guide the parallel feature registration process to specifically activate task-relevant

features. This is in contrast to the FIT pipeline in which the feature registration process is purely bottom-up and therefore task-agnostic.

In computer vision, methods that incorporate some form of visual attention do so for performance reasons rather than for simulating human cognition. Nevertheless, the general pipeline used attention-based vision models closely resembles the theoretical frameworks of FIT and Guided Search. First, bottom-up features are used to transform the input image representation, creating the feature map. Next, salient regions are identified within the feature map based on a task-dependent metric (e.g. likelihood of being an object for object detection). Finally, a second-stage processes the information from the salient regions, to complete the inference. We direct interested readers to Frintrop *et al* [12] for a more detailed overview of computational visual attention.

In this section, we will look at several forms of visual attention in computer vision. We will first look at region proposals, which model object saliency for the object detection task. Next, we look at part discovery and discriminative patches, which identify salient visual patterns for many recognition tasks. Finally, we include a brief overview of soft-attention networks.

2.1.1 Region Proposals

Region proposal methods identify regions within an image with the goal of capturing all objects within the image in as few proposals as possible. They specifically model a form of visual saliency for object detection, directing the detector where to evaluate in the image as efficiently as possible. Popular methods include Objectness [13], Category Independent Object Proposals [14], Selective Search [15], Edge Boxes [5], CPMC [16], and RPN [17]. An example of where this would be extremely beneficial can be seen in figure 2.1. An exhaustive sliding window approach would need to run the full model on the image at all locations and scales – a process that may be prohibitively slow and expensive for large models. Preprocessing with a region-proposal method is much cheaper, relying only on low-level image cues, and directly returns a manageable selection of image regions at the appropriate locations and scales.

Earlier methods [14, 15, 16] approached the proposal task as a multiple seg-

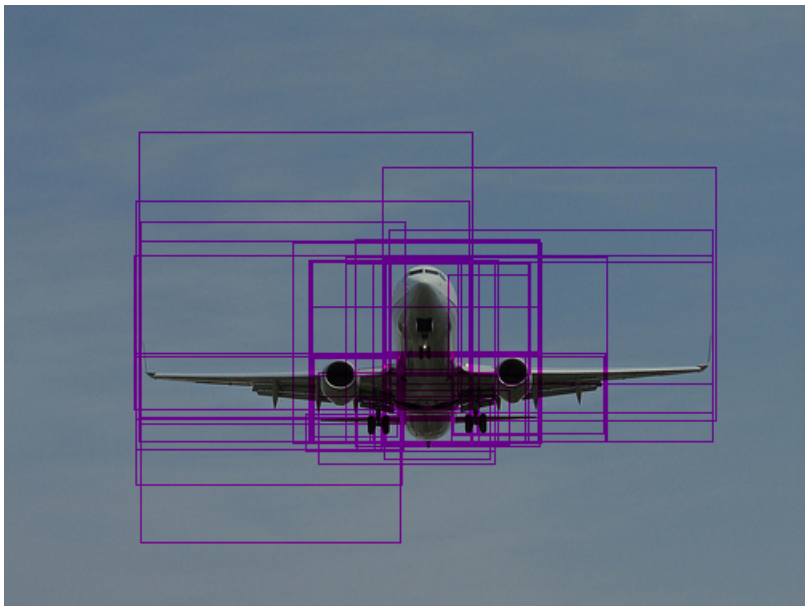

Figure 2.1: Top salient Edge Boxes [5] on an example image. Region proposals allow us to avoid unecessarily process uninteresting background regions

mentation problem, proposing multiple possible object segementations and ranking them by likelihood of being an object. Methods such as Objectness [13] and Edge Boxes [5] avoid producing full segmentations to reduce computation, as object detection methods tend to operate on entire bounding boxes. Specifically, Objectness directly scores boxes by looking for high color-contrast with the exterior of the box, low levels of superpixel straddling, and high edge density nearing the borders. Edge Boxes similarly looks for edge contours contained wholly within a bounding box, using the Structured Edge detector from Dollár *et al* [18] to generate edge maps.

In recent years, focus has shifted to unified deep network frameworks in which region proposing and the object detection task can be trained simultaneously within the same architecture. Examples include OverFeat [19], YOLO [20], Faster RCNN [17], and SSD [21]. In these frameworks, predefined anchor boxes are exhaustively generated at multiple scales, aspect ratios, and locations. The deep-network architectures then predict offsets for the box coordinates to reshape the boxes to overlapping objects based on feature maps generated from several convolutional layers.

2.1.2 Part Discovery Pre-CNN

As previously discussed, one important application of visual attention involves identifying discriminative components for the model to focus on. For example, the eyes, nose, and mouth would be some of the more discriminative and informative parts of the face as compared to a random patch of skin on the forehead or the cheek. However, identifying the discriminative visual patterns and components of a new object is not always straightforward. Not only does one want patterns that are discriminative of the category that they represent, these patterns should be easily identifiable and diverse with respect to each other to increase coverage over the example space.

Automatic part discovery for object and scene recognition is an important area of research, largely due to the difficulty of defining an appearance model. The identification of discriminative parts and patches greatly simplifies the task of object and scene recognition. Instead of focusing on the entire input frame, we now focus on detecting a set of smaller and less varied image patches that are strongly indicative of a target category.

Previous works largely focus on identifying discriminative parts or patches for training detectors using HOG representations [22]. The deformable part model [23], automatically determines the high resolution parts of an object category by first training a coarse whole-object HOG filter and then greedily partitioning areas as parts based on the magnitude of filter weights within the area. In Juneja *et al* [24] as well as in our previous Boosted Collections of Parts work [25, 26], a large number of part detectors are quickly initialized by training on a single positive patch using exemplar SVMs [27] or the faster exemplar LDA [28], then a discriminative and diverse subset of the detectors is identified and iteratively refined by mining for additional positive patches from the training set. Singh *et al* [29] initializes part detectors by clustering similar patches and iteratively refines across two splits of the training data. Their selection criteria focuses on cluster purity and discriminativeness. Doersch *et al* [30] explores a similar criteria, but uses an extension of mean-shift mode-seeking over density ratios of positive and negative data to identify discriminative parts. Sun and Ponce [31] also initialize part detectors by clustering patches, but they enforce diverse part selection using a sparse regularization term.

2.1.3 Automated Part Discovery in CNNs

Convolutional Neural Networks (CNNs) refer to a family of neural network architectures with the explicit property of being shift-invariant. On image data, a CNN architecture typically contains a series of convolutional layers applied on image data (or any tensor with height, width, and depth). Each convolutional layer can be seen a sliding window filter that applies a linear transformation on the elements within its current window as it strides across the width and height dimensions. For example, let x be an input image of dimensions $H \times W \times D$. When the filter's top left corner is located at $x_{i,j}$, it outputs the following K'-dimensional vector:

$$y_{i''',j''',k'} = \sum_{i''=1}^{M} \sum_{j''=1}^{H} \sum_{k=1}^{D} w_{i'',j'',k} \times x_{i+i''-1,j+j''-1,k} \text{ for } k' = 1...K' \qquad (2.1)$$

The full output after convolving over the entire input is an $H' \times W' \times K'$ feature map. Here, H' and W' are determined by the horizontal and vertical strides of the convolutional filter as it slides across the input x. As the output is another tensor with width, height, and depth, we can easily chain a series of convolutional layers, leading to "deep" architectures. It is worth noting that we can reduce a convolutional layer to a traditional multi-layer perceptron layer (fully-connected layer) by setting the window's width and height to exactly match that of its input.

CNNs have drastically improved benchmark performance on many traditional computer vision tasks. One of the earliest successful applications of CNNs was LeCun *et al* [32] in optical classification, in which the authors propose LeNet-5, a 7-layer CNN to recognize hand-written digits. More recently, Krizhevsky *et al* [33] popularized the AlexNet architecture which made significant improvements over existing methods in the large-scale image classification challenge ImageNet [34]. Following the success of AlexNet, more accurate CNN architectures have been proposed, including VGG [35], Inception [36], and ResNet [37]. The feature representations of these networks, after being pre-trained on the ImageNet classification challenge, have been shown to be very effective in nearly all related computer vision tasks, including but not limited to scene recognition [38], object detection [39, 40], and semantic/instance segmentation [39, 41, 42].

The CNN shares some important similarities with previous approaches.

As noted in LeCun *et al*[32], the stacking of convolutional layers with subsampling every few layers "ensures some degree of shift, scale, and distortion invariance." This is similar to the previously popular HOG feature pyramids generated from running HOG filters at multiple scales of the image. One can think of CNN filters as much more expressive HOG filters that are end-to-end-trainable with the main task. While end-to-end training will not always outperform a compositional approach (training the model one sub-problem at a time), it benefits from being much easier to setup and learns its own internal representations that are importantly jointly optimized for the task with the rest of the architecture's components.

An interesting result of training CNNs end-to-end is that the filters will naturally learn to detect patterns that benefit the main task – arguably a form of automatic part discovery. CNN visualization works such as Zeiler *et al* [43] suggest that the model starts by learning low-level edge-like cues at the bottom and becomes increasingly abstract as layers are stacked on. From the bottom up, low level edge filters are pooled to create various shape filters. These are are combined to form simple parts such as wheels and eyes – parts which are further pooled to capture entire viewpoints of vehicles or faces of animals. With the automatic representation learning and part-discovery due to end-to-end CNN training, it is no longer necessary to manually engineer feature representations or to manually identify salient image patterns from which to train part-detectors. Further, the parts and patterns determined by the CNNs training may be better choices than manually engineered solutions (given sufficient data), as their selection was driven directly by the model's task performance as opposed to human intuition. Our work in latent attention will exploit similar behavior in end-to-end training of neural network architectures, allowing the model to self-identify task-relevant image regions.

2.1.4 Soft Attention Networks

Up until now, we have looked at examples of visual attention in which the tasks's objective is well-known beforehand. We now look at a more general framework for visual attention in which the attention behavior may be adapted to a different objective on the fly.

We begin by defining the soft attention mechanism. Soft attention here

refers to a soft, differentiable alternative to the argmax selection:

$$\hat{v} = \arg\max_{v_i \in V} s(v_i) \tag{2.2}$$

where we wish to select the vector $\hat{v}$ from a set of N vectors $v_i \in V$ based on their respective scores $s(v_i)$. As this is non-differentiable, we approximate the hard-selection with a weighted average:

$$\hat{v} = \sum_{i \in N} g(s(v_i)) v_i \text{ s.t. } \sum_{i \in N} g(s(v_i)) = 1 \tag{2.3}$$

Here, $g(s(v_i))$ is the normalization function over selection scores. It is most commonly a softmax distribution over all vectors $v_i \in V$:

$$g(s(v_i)) = \frac{\exp s(v_i)}{\sum_{j \in N} \exp s(v_j)} \tag{2.4}$$

Note that as the softmax distribution approaches 1-hot, soft-attention approaches argmax selection.

Soft attention has seen applications in numerous deep network architectures to tackle various tasks. Bahdanau *et al* [44] uses soft attention as a soft alignment between a source sentence and its target translation. Xu *et al* [45] similarly uses this technique to align different parts of the image with the next word to predict in an image captioning framework. Sukhbaatar *et al* [46] learns to predict a soft distribution over a set of previously made statements to respond to a natural language query.

The significance of soft-attention to our work is the abstraction of the scoring function $s(v_i)$. Specifically, suppose we parameterized the scoring function as $s(v_i, \theta)$, then we can adjust the visual attention behavior on the fly by predicting the appropriate θ. We address this in our chapters on attention for visual question answering, in which we try to vary the behavior of visual attention for different questions about the image.

2.2 Vision Tasks

The field of computer vision spans a diverse range of tasks requiring some form of visual perception. In our work, we focus on incorporating the ability

to learn task-driven visual attention in three main tasks: keypoint localization, fine-grained image recognition, and visual question answering. We provide an overview of each of the tasks in the following section.

2.2.1 Keypoint Localization and Regression in CNNs

In the following work, we refer to the task of localizing annotated pixel-locations (eg. center of the eye or nose) as keypoint localization. The keypoint/part localization task is strongly related to object detection in that it was used to model object detectors capable of capturing various poses. The use of pose in object detection can be seen in the line of work deriving from the Pictorial Structure models [47, 48], in which recognition was modeled as localizing rigid parts arranged in a deformable configuration. Popular datasets for keypoint localization include Leeds Sports [49, 50], Poselets on Pascal [51], UCSD birds [52], and more recently MSCOCO [53].

Due to the recent advancements in deep learning, keypoint localization methods have shifted from classical approaches that focus on localizing various part-based templates ([54, 55, 56]) to models based on end-to-end trained CNNs. Most relevant to our work are CNN architectures that attempt to regress to the target coordinates. Prior to our work, the most notable applications of of deep regression networks to keypoint localization are Toshev *et al* [57] and Sun *et al* [58], which use cascades of deep network based regressors for human pose estimation and facial keypoint localization respectively. At each stage of the cascades, the network uses a region around the previous prediction to acquire higher resolution inputs. This allows the models to slowly adjust their prediction context in a coarse-to-fine fashion. The cascade addresses the problem in which CNNs expect a fixed-size input – feeding in the entire image will require downsampling, whereas feeding in smaller regions of the image would involve knowing where to crop and how much context is necessary. Instead of cascades, our work as described in Chapter 3 relies on multiple regions sampled with Edge Boxes from the image and simultaneously predicts all keypoints. Varying sized regions provide varying resolution and context, and we achieve more robust predictions from multiple regions with statistical outlier removal.

2.2.2 Fine-Grained Image Recognition

Fine-grained visual recognition refers to classification between visually similar and closely related categories. Differences may be as minute as feather color or beak shape between birds [52, 59], petal shapes between various plants [60], or even fur patterns between various types of dogs [61]. Prior work in this field focuses on localizing informative parts of objects and then extracting features from them for classification. Using pairs of localized keypoints, Berg *et al* [62] learn a set of highly discriminative features for fine-grained classification. Farrell *et al* [63] and Branson *et al* [64] use pose normalized representations of birds and their regions (head, torso, entire bird) followed by feature extraction for classification. Liu *et al* [1] extend the exemplar based model of [65] with pose information for keypoint localization and subsequent classification of birds. Based on the very successful framework of the RCNN [66], Zhang *et al* [67] perform bird classification using three localized bird regions: head, torso, and full body.

The above mentioned methods are highly dependent on accurate keypoint and bird region localization. In fact, [62, 63] rely on the groundtruth bird bounding box at test time to localize keypoints and to perform classification. Our work overcomes this bottleneck of localization and we demonstrate state-of-the-art classification performance using the framework of [67] along with our localized regions.

2.2.3 Visual Question Answering

Visual question answering (VQA) is the task of answering a natural language question about an image. VQA includes many challenges in language representation and grounding, recognition, common sense reasoning, and specialized tasks such as counting objects and reading signs. To some degree, VQA benchmarks were proposed as a vision-language task with a less ambiguous evaluation than one such as image-captioning. It is much easier to identify correct and incorrect responses to a question about an image than to determine whether a random caption in a dataset is a valid match for an image. Further, models tackling various vision-language tasks are often similar in that they contain a mechanism for comparing vision and language feature representations. As such, a improvements in the VQA task will likely

transfer to other related vision-language tasks as well.

Our work experiments on the VQA dataset of Antol *et al* [6] due to the open ended nature of its question and answer annotations. Questions are collected by asking annotators to pose difficult problems for a smart robot, and multiple answers are collected for each question. We experiment on the multiple-choice setting as its evaluation is less ambiguous than that of open-ended response evaluation. Most other visual question answering datasets [68, 69] are based on reformulating existing object annotations into questions, which provides an interesting visual task but limits the scope of visual and abstract knowledge required. Accompanying approaches tend to use recurrent networks to model language and predict answers [68, 6, 69, 70]. We find a fixed-length representation for vision and language to be highly effective, and our approach differs at a high level in our focus on learning where to look. Simple Bag-Of-Words models have been shown to perform roughly as well if not better than sequence-based LSTM[68, 6]. Further, Yu et al. [69] propose a Visual Madlibs dataset for fill-in-the-blank and question answering and focus their approach on learning latent embeddings, finding normalized CCA [71] to outperform recurrent networks for embedding.

CHAPTER 3

LOCALIZING KEYPOINTS

The most common approach to keypoint localization is to learn a set of keypoint detectors to model appearance and an associated spatial model [67, 2, 1, 64] to capture their spatial relations. The keypoint detectors generate a set of likely candidates per part and a spatial model is used to infer the most likely configuration. Keypoint detectors typically model local appearance and thus an approach has to rely on expressive spatial models to capture long range dependencies. Alternatively, the keypoint detectors can condition their predictions on larger spatial support and jointly predict several keypoints [72], reducing the need to explicitly model inter-keypoint relationships.

In this chapter, we describe a method for learning a keypoint localization model that relies on larger spatial support to jointly localize several keypoints and predict their respective visibilities. Leveraging recent developments in Convolutional Neural Networks (CNNs), we introduce a framework that outperforms the state-of-the-art for localizing bird keypoints for eyes, beaks, etc. on the CUB dataset. Further, while CNN-based methods suffer from a loss of image resolution due to the fixed-sized inputs of the networks, we introduce a simple sampling scheme that allows us to work around the issue without the need to train cascades of coarse-to-fine localization networks [57, 58].

Our approach to keypoint localization mainly draws inspiration from the use of regression in networks in the MultiBox approach by Erhan *et al* [73]. The authors train a deep network which regresses a small number of bounding boxes ($\sim$ 100) as object bounding box proposals, along with a confidence value for each bounding box.

Our work is applied to the Caltech-UCSD Birds dataset. The most closely related work on that dataset is from Liu *et al* [1, 2]. Their works achieve remarkable performance on both keypoint localization and visibility prediction using ensembles of pose exemplars and part-pair detectors. We compare our performance with theirs using metrics defined in their work.

3.1 Method

We design our model to simultaneously predict keypoint locations and their visibilities for a given image patch. To share the information across categories, our model is trained in a category agnostic manner. At test time, we efficiently sample each image with Edge Boxes, make predictions from each Edge Box, and reach a consensus by thresholding for visibility and reporting the medoid.

3.1.1 Training Convolutional Neural Networks for Keypoint Regression

Our network is based on AlexNet ([33]), but modified to simultaneously predict all keypoint locations and their visibilities for any given image patch. AlexNet is an architecture with 5 convolutional layers and 3 fully connected layers. Henceforth, we refer to the 3 fully connected layers as fc6, fc7, and fc8. We replace the final fc8 layer with two separate output layers for keypoint localization and visibility respectively. Our network is trained on Edge Box ([5]) crops extracted from each image and is initialized with a pre-trained AlexNet ([33]) trained on the ImageNet ([34]) dataset. Each Edge Box is warped to 227×227 pixels before it can be fed through the network. We apply padding to each Edge Box such that the warped 227×227 pixel crop has 16 pixels of buffer in each direction.

Given N keypoints of interest, we train a network to output an N dimensional vector $\hat{v}$ and a $2N$ dimensional vector $\hat{l}$ corresponding to the visibility and location estimates of each of the keypoints k_i, $i \in \{1, N\}$, respectively. The corresponding groundtruth targets during training are v and l. We define v to consist of indicator variables $v_i \in \{0, 1\}$ such that $v_i = 1$ if keypoint k_i is visible in the given Edge Box image before padding is performed, and 0 otherwise. The groundtruth location vector l is of length $2N$ and consists of pairs (l_{x_i}, l_{y_i}) which are the normalized $(\tilde{x}, \tilde{y})$ coordinates of keypoint k_i with respect to the un-padded Edge Box image. Output predicted from the network, $\hat{v}_i \in [0, 1]$, acts as a measure of confidence of keypoint visibility, and 2D locations predicted by the network are denoted by $\hat{l}_i$.

We use the *Caffe* framework ([74]) for training our deep networks. To train a network optimized for both tasks simultaneously, we define our losses as

follows:

$$\mathcal{L}_{vis} = ||v - \hat{v}||_2^2 \quad \text{and} \quad \mathcal{L}_{loc} = \sum_{i=1}^{N} v_i \cdot \left[(l_{x_i} - \hat{l}_{x_i})^2 + (l_{y_i} - \hat{l}_{y_i})^2 \right] \tag{3.1}$$

$$\mathcal{L}_{net} = \mathcal{L}_{vis} + \mathcal{L}_{loc} \tag{3.2}$$

The visibility loss $\mathcal{L}_{vis}$ is the squared Euclidean distance between the ground truth visibility label vector v, and the predicted visibility vector $\hat{v}$. The values in our $\hat{v}$'s always lie between 0 and 1 as they are obtained after squashing network outputs with a sigmoid function. The keypoint localization loss $\mathcal{L}_{loc}$ is a modified Euclidean loss, in which we set the loss between the prediction and the target to be 0 if $v_i = 0$ i.e. if the keypoint k_i is absent in the given image. The final training loss ($\mathcal{L}_{net}$) is given by the sum of the two losses.

To construct our training set for predicting keypoint visibility and locations, we extract up to 3000 Edge Boxes per image. To train a robust predictor, we need a collection of training images with high variability in which different subsets of keypoints are visible. We generate examples that satisfy this criteria by retaining a subset of Edge Boxes which have at least 50% of their area contained inside the groundtruth bounding box and have at least 20% intersection over union overlap (IOU) with the groundtruth bounding box. We also included up to 50 random boxes per image from outside the bounding box as negative background examples. We augment our dataset with left/right flips. After flipping, appropriate changes were applied to the label vectors. This consisted of swapping orientation-sensitive keypoints such as "left eye" and "left wing" with "right eye" and "right wing", and updating their respective coordinates and visibility indicators. We first train our model on 25 images per class and tune algorithmic and learning rate parameters on a held-out validation set comprising the remaining 4-5 images per class. Finally, we re-train using the entire training set before running our model on the test set.

3.1.2 Combining Multiple Keypoint Predictions

Our algorithm for dealing with predictions from multiple Edge Boxes at test time is illustrated in Fig. 3.1. Due to the variance from making predictions

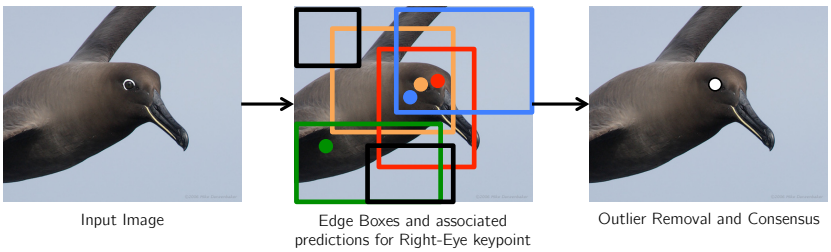

Figure 3.1: The pipeline of our keypoint localization process: Given an input image, we extract multiple edge boxes. Using each edge box, we make predictions for the location of each of the 15 keypoints, along with their visibility confidences. We then find the best predicted location by performing confidence thresholding and finding the medoid. The process is illustrated for the right eye keypoint (Black edge boxes without associated dots make predictions with confidences below the set threshold, and green is an outlier with a high confidence score).

from multiple unique subcrops of the image, we need to form a consensus from the multiple predictions. In our experiments, we found that after removing predictions with low visibility confidences, the remaining predictions had a peaky distribution around the ground truth. We use medoid as a robust estimator for this peak and found it to be effective in most cases (Fig. 3.2). For the task of localizing part regions around keypoints (described in chapter 4), we found on our train/val split that we achieved better localization performance if we kept a set of good predictions (referred to as *inliers*) instead of using only the medoid. We now describe our procedure for obtaining a tight set of inliers and our choice of parameters. For the keypoint prediction task, we only use the visibility thresholds and report the medoid.

Case 1: Ground Truth Object Box Given:

We first describe our method in the case that the ground truth object boxes are given. Using the ground truth object box, we retain the generated Edge Boxes that are mostly contained within and have an IOU of at least 0.2. This results in roughly 50-200 remaining Edge Box subcrops per image. Each subcrop is then independently fed through our keypoint prediction network, returning a set of normalized keypoint predictions and visibilities.

Because each subcrop is expected to cover less than the whole object and contain only a subset of the keypoint predictions, we drop any prediction if

its corresponding visibility is below 0.6. Because we make use of multiple overlapping subcrops, it is very likely that at least one of them will lead to a prediction with a sufficiently high visibility score, thereby allowing us to be much more aggressive with the false positive filtering.

Given multiple remaining keypoint predictions per keypoint with sufficiently high visibility scores, we then proceed to remove outliers. To do so, we threshold on a modified Z-score based on a description given by Iglewicz and Hoaglin ([75]). The modified Z-score is one that is re-defined using medoids and medians in place of means, as the former estimates are more robust to outliers.

Let p_i where $i = 1, \cdots, M$ be the set of M surviving un-normalized keypoint predictions (for a given keypoint) in (x, y) image coordinates. We first define $\bar{p}$ to be the medoid prediction such that:

$$\bar{p} = \arg\min_{p_j} \sum_{i=1}^{M} ||p_j - p_i||_2, \ j \in \{1, ..., M\} \tag{3.3}$$

In other words, $\bar{p}$ is the prediction such that its Euclidean distance from all other predictions for that keypoint is minimal. While this optimization is costly at a large scale, we typically deal with only 10-20 predictions at a time after thresholding for visibility scores. To compute the modified Z-score we use:

$$Z_i = \frac{\lambda \ ||p_i - \bar{p}||_2}{\text{median } (||p_i - \bar{p}||_2)}, \ i \in \{1, ..., M\} \tag{3.4}$$

Here, the denominator is the median absolute deviation, or simply the median distance from the medoid $\bar{p}$. We use the recommended $\lambda = 0.6745$. The above procedure is separately computed for all 15 sets of keypoint prediction candidates. Finally, we drop any keypoint prediction with $Z_i > 0.35$, a threshold that was experimentally determined on the held-out set.

Case 2: Ground Truth Object Box Not Given:

Our ground truth object box not given scenario requires little change from the above case. Using the Edge Box ranking, we found that most of our "good" Edge Boxes fell within the top 600 Edge Boxes per image, saving us a lot of computation. Tuning parameters on our train/val split, we found

that an even more aggressive visibility threshold of 0.94 and a Z-score threshold of 0.3 gave the best results.

Medoid-Shift:

While the simple Z-score thresholding combined with the medoid achieves excellent results, as we will demonstrate in the results section, we were able to further improve our results by using medoid-shifts ([76]). We use the medoid of the largest output cluster from the algorithm instead of the medoid computed over all the visibility-filtered predictions.

3.2 Results

We evaluate our keypoint prediction model on the Caltech UCSD-Birds dataset by Wah *et al.* This dataset contains 200 bird categories with 15 keypoint location and visibility labels for each of the total of 11788 images. We first evaluate our keypoint localization and visibility predictions against other top-performing methods.

3.2.1 Keypoint Localization and Visibility Prediction

Table 3.1 reports our keypoint and visibility performance without using any ground truth bounding box information. Our medoid method reports the medoid of predictions above a visibility threshold, as seen in the red star in Fig. 3.2. Our "mdshift" method reports the new medoid computed using medoid-shift, which is the blue circle in Fig. 3.2. We used the evaluation code provided by the authors of [1] to measure our performance using the metrics defined in their work. In short, PCP (Percent Correct Parts) is the percentage of keypoints localized within 1.5 times the annotator standard deviation. We received the pre-computed standard deviatons and evaluation code from the authors of [1] to avoid any discrepancies during evaluation. AE (Average Error) is the mean euclidean prediction error, capped at 5 pixels, computed across examples where a prediction was made and a ground truth location exists. FVR and FIR refer to False Visibility Rate and False Invisibility Rate

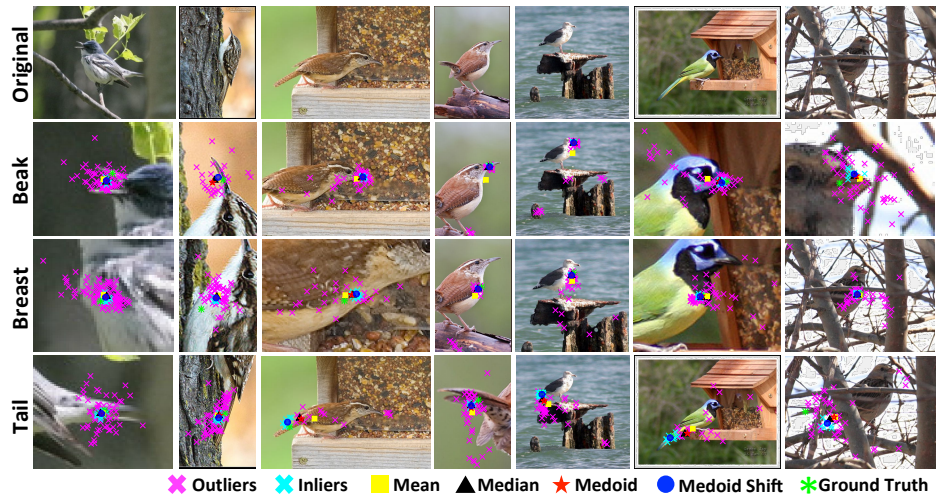

Figure 3.2: Qualitative results for a subset of the keypoints. Predictions for most of the images cluster tightly. Therefore, simple prediction methods such as medoids work well. Medoid shift adds to the robustness, leading to further improvements (second last column). Primary failure mode is when visibility thresholding fails to rule out clusters of false positives (bottom right).

Method	PCP	AE	FVR	FIR
Poselets ([77])	24.47	2.89	47.9	17.15
Consensus ([65])	48.70	2.13	43.9	6.72
Exemplar ([1])	59.74	1.80	28.48	**4.52**
Ours (medoid)	68.7	1.4	**17.1**	5.2
Ours (mdshift)	**69.1**	**1.39**	**17.1**	5.2
Human ([1])	84.72	1.00	20.72	6.03

Table 3.1: Localization and Visibility Prediction Performance of various methods without using the ground truth Bounding Box

respectively. The additional methods for comparison are the same as listed in their paper.

Compared to the top-performing methods that also predict visibility, our method achieves the best numbers in three out of four metrics. Our PCP and AE metrics outperform other methods in the table, with our medoid-shift variant performing slightly better. Our FIR is higher because we are using the visibility threshold tuned on the part-localization task. A slightly lowered threshold would lower the FIR and raise the FVR without significantly affecting the PCP.

The highest reported PCP is 66.7% due to [2], which also predicts visibilities but did not report them. We compare against their PCP in Table 3.2.

PCP	Ba	Bk	Be	Br	Cr	Fh	Ey	Le	Wi	Na	Ta	Th	Total
Liu '13	62.1	49.0	69.0	67.0	72.9	58.5	55.7	40.7	71.6	70.8	40.2	70.8	59.7
Liu '14	64.5	**61.2**	71.7	70.5	76.8	**72.0**	**70.0**	45.0	74.4	**79.3**	46.2	**80.0**	66.7
Ours	**74.9**	51.8	**81.8**	**77.8**	**77.7**	67.5	61.3	**52.9**	**81.3**	76.1	**59.2**	78.7	**69.1**

Table 3.2: Comparison of per-part PCP with Liu *et al* 2013 [1] and Liu *et al* 2014 [2]. The abbreviated part names from left to right stand for back, beak, belly, breast, crown, forehead, eye, leg, wing, nape, tail, and throat.

Because our method differs significantly from theirs, we outperform them in only 7 of the listed part categories despite having a better overall PCP, suggesting further improvements by targeting the differences in our models' behaviors.

3.3 Conclusion

We presented a method for obtaining state-of-the-art keypoint predictions on the Caltech UCSD-Birds dataset. We demonstrated that conditioning the predictions on multiple object proposals for sufficient image support can reliably improve localization predictions without using a cascade of coarse-to-fine networks. We tackle the problem of fixed-size inputs when using neural networks by sampling predictions from several boxes and determining the "peak" of the predictions with medoids. In the next chapter, we will look at applying these keypoint predictions to part-aligned fine-grained image classification.

CHAPTER 4

FINE GRAINED CLASSIFICATION WITH ALIGNED PARTS

Fine-grained image categorization is the task of accurately separating categories where the distinguishing features may be as minute as a different fur pattern, shorter horns, or a smaller beak. The widely accepted and popular approach of dealing with such a task is intuitive: align analogous regions and compare. The alignment process allows the model to compare apples to apples, and oranges to oranges. A specific set of parameters can focus exclusively on learning the minute differences between beak shapes whereas a different set can focus on wing patterns. In this chapter, we describe how we use our keypoint prediction results from the previous chapter to conduct region-aligned classification.

4.1 From Keypoints to Regions

In order to align analogous regions to perform fine-grained classification, we must first map our pixel-level keypoint predictions to alignable image regions from which we can extract features. To do this, we first use the keypoint mapping as used in works by Zhang *et al* [67, 78]. Using the keypoints, three regions are identified from each bird: head, torso, and whole body. The head is defined as the tightest box surround the beak, crown, forehead, eyes, nape, and throat. Similarly, the torso is the box around the back, breast, wings, tail, throat, belly, and legs. The whole body bounding box is the object bounding box provided in the annotations.

To handle the case when ground truth bounding box is not given at test time, we use an overlap heuristic based on the predicted head and torso boxes. We first start by finding the tightest box around the predicted head and torso boxes. While this initial box will do well for birds in their canonical poses, it will result in an undersized box in many cases because the keypoints

	Method	Head	Torso	Whole Body
GT Bbox	Part-Based RCNN ([67])	68.2	79.8	N/A
	Deep LAC ([79])	74.0	**96.0**	N/A
	Ours (single GT bbox)	75.6	90.2	N/A
	Ours (multiple)	88.8	93.9	N/A
	Ours (multiple, mdshift)	**88.9**	94.3	N/A
No GT Bbox	Part-Based RCNN ([67])	61.4	70.7	**88.3**
	Exemplar ([1])	79.9	78.3	N/A
	Ours (multiple)	87.8	**89.0**	84.5
	Ours (multiple, mdshift)	**88.0**	88.7	84.6

Table 4.1: Comparison of Part Localization Performance: Our method based on keypoint prediction from Edge Boxes shows significant improvement over previous work.

do not always capture the full extent of the bird. We then assume that there exists an Edge Box with a high edge score that better captures the whole bird. To let the box expand to capture more of the object, we first identify the Edge Boxes such that the tightest box is at least 90% contained within and has at least 50% IOU overlap. The final whole body bounding box is the Edge Box that passes both criteria that also has the highest Edge Box object score. If no Edge Box passes the overlap test, we fall back to the starting tightest box.

The results in Table 4.1 demonstrate that our keypoint predictions are useful in generating accurate part boxes. Our lower performing single GT Bbox method suggests that our use of multiple predictions from Edge Boxes allows for more accurate predictions. Further, we also computed head and torso boxes using the keypoint predictions from [1] as shown in the "Exemplar" row. Based on their accuracy, their boxes should also be able to improve the results of [67].

Next, given bounding boxes for head, torso, and whole body, we use the same SVM-classification framework as used by [67] to conduct part-aligned fine-grained classification. Specifically, AlexNet fc6 features are extracted from each of the localized regions, then concatenated into a feature vector of length 4096×3 and used for 200-way linear 1-vs-all SVM classification.

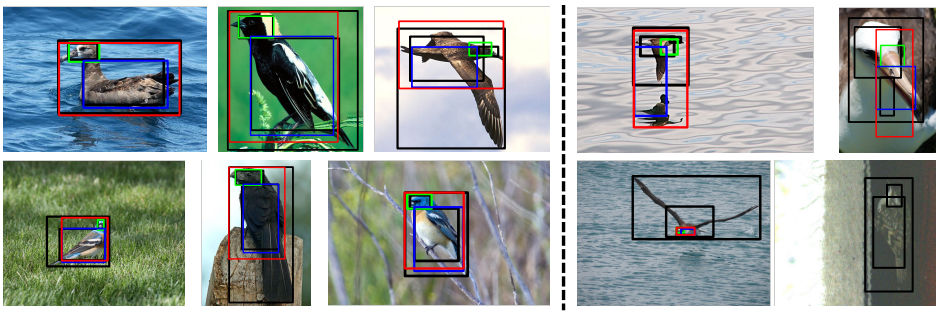

Figure 4.1: Examples of good (left) and failed (right) localization results: The ground truth boxes are in solid black. The head, torso, and whole body boxes are in green, blue and red respectively. The head is correctly localized in most of the above examples. In the top row middle example, even though the whole body box IOU is low, most of the missed area is actually background due to the bird extending its wings. In the bad examples, we show that we mostly fail in rare close-ups and when there are multiple instances.

4.2 Fine-Grained Classification

We now test our part-predictions in a fine-grained classification setting. These results are shown in Table 4.2. To do this, we train three networks to re-implement the three-part framework of [67] as described in the previous section. The oracle performance refers to the classification assuming ground truth keypoints at test time. While [67] reports an oracle accuracy of 82.0%, we compare with the highest we were able to achieve with our implementation: 81.5%. This is likely due to minor differences in network training parameters. We also tried both fc6 and fc7 features and found that fc6 performed a little better. Although [67] and [64] noted that their drops in accuracy from using ground truth parts to predicted parts were surprisingly small, our relative improvements suggest that it is still worthwhile to focus on better localization. Further, we perform at least as well as the contemporary Deep LAC model ([79]), likely due to our better localization of the more discriminative head regions.

In Fig. 4.2, we show how our accuracy is affected from the ground truth keypoint ideal case (Oracle) to the use of predicted keypoints (GT Bbox), and finally with the GT Bbox removed (No GT Bbox). Unsurprisingly, the better localization at test time allows for a significantly smaller drop as annotations are removed.

The same plot also shows an ablation test of individual parts. It appears that the bulk of our performance comes from discriminating localized bird

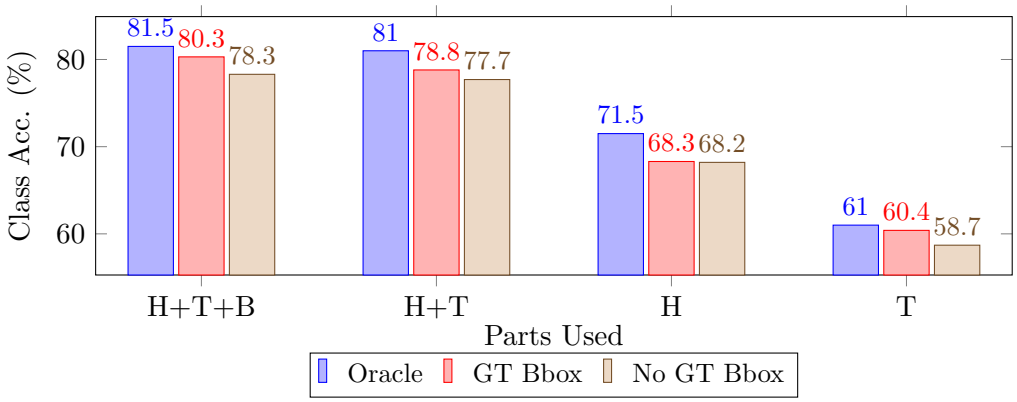

Figure 4.2: Camporison of classification accuracies obtained using varying combinations of parts localized under different conditions

	Method	Acc.
Oracle	Oracle Parts + SVM	**81.5**
GT Bbox	DPD ([78])	51.0
	Symbiotic ([80])	59.4
	Alignment ([81])	62.7
	DeCAF ([82])	65.0
	POOF ([62])	56.8
	Part-Based RCNN ([67])	76.4
	Deep LAC ([79])	80.3
	Ours (mult, medoid)	80.3
	Ours (mult, mdshft)	**80.3**
No GT Bbox	Pose Norm ([64])	75.7
	Part-Based RCNN ([67])	73.9
	Ours (mult, medoid)	78.2
	Ours (mult, mdshft)	**78.3**

Table 4.2: Comparison of our classification with other works

heads. This is also supported by [64] which observed that of their learned poses, the one that corresponded to the head was the most discriminative. This suggests that most of our improvement over our base method of [67] comes from significantly improving our head part localization (shown in Table 4.1).

4.3 Conclusion

We presented an extension of our keypoint prediction work to fine-grained classification. We demonstrated the importance of keypoint prediction with

accurate visibility prediction in robustly localizing image regions. Using our part-localization approach, we improved upon existing work in both localizing head and torso regions, and subsequently the overall classification accuracy.

CHAPTER 5

LATENT ATTENTION FOR VISUAL QUESTION ANSWERING

5.1 Introduction

Visual question answering (VQA) is the task of answering a natural language question about an image. VQA includes many challenges in language representation and grounding, recognition, common sense reasoning, and specialized tasks like counting and reading. In this paper, we focus on a key problem for VQA and other visual reasoning tasks: knowing where to look. Consider Figure 5.1. It is easy to answer "What color is the walk light?" if the light bulb is localized, while answering whether it's raining may be dealt with by identifying umbrellas, puddles, or cloudy skies. We want to learn where to look to answer questions supervised by only images and question/answer pairs. For example, if we have several training examples for "What time of day is it?" or similar questions, the system should learn what kind of answer is expected and where in the image it should base its response.

Learning where to look from question-image pairs has many challenges. Questions such as "What sport is this?" might be best answered using the full image. Other questions such as "What is on the sofa?" or "What color is the woman's shirt?" require focusing on particular regions. Still others such as "What does the sign say?" or "Are the man and woman dating?" require specialized knowledge or reasoning that we do not expect to achieve. The system needs to learn to recognize objects, infer spatial relations, determine relevance, and find correspondence between natural language and visual features. Our key idea is to learn a non-linear mapping of language and visual region features into a common latent space to determine relevance. The relevant regions are then used to score a specific question-answer pairing. The latent embedding and the scoring function are learned jointly using a margin-based loss supervised solely by question-answer pairings. We per-

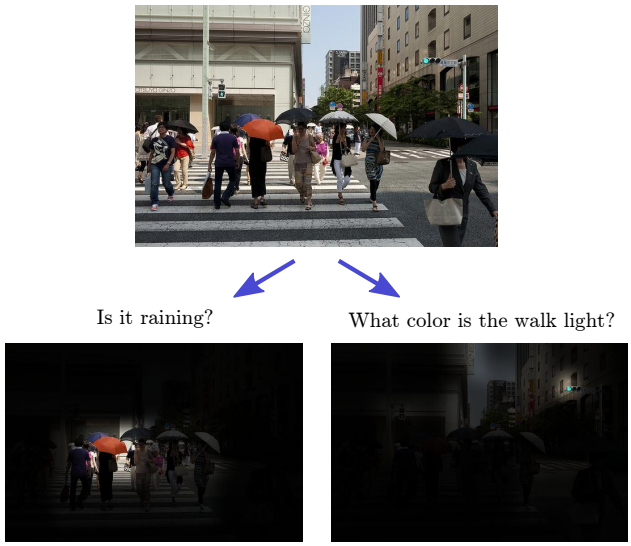

Figure 5.1: Our goal is to identify the correct answer for a natural language question, such as "What color is the walk light?" or "Is it raining?" We focus on the problem of learning where to look. The above figure shows example attention regions produced by our model.

form experiments on the VQA dataset ([6]) because it features open-ended language, with a wide variety of questions (see Figure 5.2). We focus on its multiple-choice format because its evaluation is much less ambiguous than open-ended answer verification.

We focus on learning where to look and provide useful baselines and analysis for the task as a whole. Our contributions are as follows:

- We present an image-region selection mechanism that learns to identify image regions relevant to questions.
- We present a learning framework for solving multiple-choice visual QA

Figure 5.2: Examples from VQA ([6]). From left to right, the above examples require focused region information to pinpoint the dots, whole image information to determine the weather, and abstract knowledge regarding relationships between children and stuffed animals.

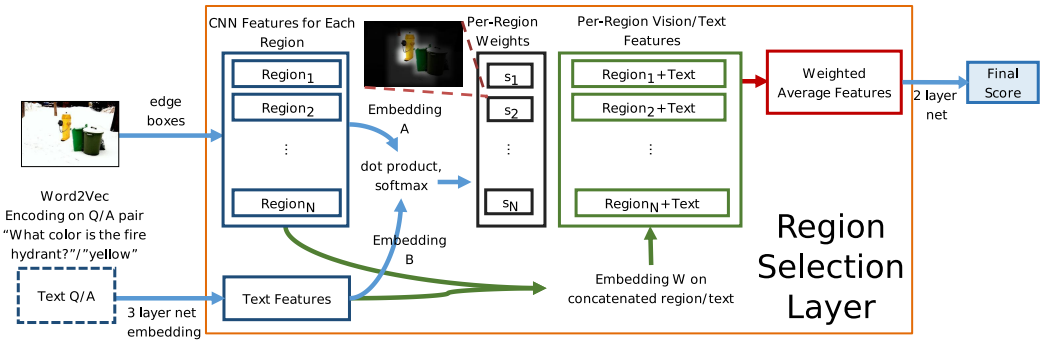

Figure 5.3: Overview of our network for the example question-answer pairing: "What color is the fire hydrant? Yellow." Question and answer representations are concatenated, fed through the network, then combined with selectively weighted image region features to produce a score.

with a margin-based loss that significantly outperforms provided baselines from [6].

- We provide a detailed comparison with various baselines to highlight exactly when our region selection model improves VQA performance

5.2 Approach

Our method learns to embed the textual question and the set of visual image regions into a latent space where the inner product yields a relevance weighting for each region. See Figure 5.3 for an overview. The input is a question, potential answer, and image features from a set of automatically selected candidate regions. We encode the parsed question and answer using word2vec ([83]) and a three-layer network. Visual features for each region are encoded using the top two layers (including the output layer) of a CNN trained on ImageNet ([84]). The language and vision features are then embedded and compared with a dot product, which is soft-maxed to produce a per-region relevance weighting. Using these weights, a weighted average of concatenated vision and language features is the input to a two-layer network that outputs a confidence for the answer candidate.

Our model is inspired by End-to-End Memory Networks [85] proposed for answering questions based on a series of sentences. The regions in our model are analogous to the sentences in theirs, and, similarly to them, we learn a linear embedding to project question and potential features into a shared

subspace to determine relevance. Our method differs in many details such as the language model and more broadly in that we are answering questions based on an image, rather than a text document.

5.2.1 QA Objective

Our model is trained for the multiple choice task of the VQA dateset. For a given question and its corresponding choices, the objective of our network aims to maximize a margin between correct and incorrect choices in a structured-learning fashion. We achieve this by using a hinge loss over predicted confidences y.

In our setting, multiple answers could be acceptable to varying degrees, as correctness is determined by the consensus of 10 annotators. For example, most may say that the color of a scarf is "blue" while a few others say "purple". To take this into account, we scale the margin by the gap in number of annotators returning the specific answer:

$$\mathcal{L}(y) = \max_{\forall n \neq p}(0, y_n + (a_p - a_n) - y_p). \tag{5.1}$$

The above objective requires that the score of the correct answer (y_p) is at least some margin above the score of the highest-scoring incorrect answer (y_n) selected from the set of incorrect choices ($n \neq p$). For example, if $\frac{6}{10}$ of the annotators answer p ($a_p = 0.6$) and 2 annotators answer n ($a_n = 0.2$), then y_p should outscore y_n by a margin of at least 0.4.

5.2.2 Region Selection Layer

Our region selection layer selectively combines incoming text features with image features from relevant regions of the image. To determine relevance, the layer first projects the image features and the text features into a shared N-dimensional space, after which an inner product is computed between each question-answer pair and all available regions.

Let $V = (\vec{v}_1, \vec{v}_2, ...\vec{v}_K)$ be a collection of visual features extracted from K image regions and $\vec{q}$ be the feature representation of the question and candidate answer pair. The forward pass to compute the relevance weighting

of the jth region is computed as follows:

$$g_j = (A\vec{v}_j + \vec{b}^A)^\top (B\vec{q} + \vec{b}^B) \tag{5.2}$$

$$s_j = \frac{e^{g_j}}{\sum_k e^{g_k}} \tag{5.3}$$

Here, vectors $\vec{b}$ represent bias vectors for each affine projection. The inner product forces the model to compute region-question relevance (g_j) in a vector similarity fashion. Using softmax-normalization across 100 regions per image ($K = 100$) gives us a 100-dimensional vector $\vec{s}$ of normalized relevance weights.

The vector $\vec{s}$ is then used to compute a weighted average across all region features. We first construct a language-vision feature representation for each region by defining $\vec{d}_j$ as the concatenation of $\vec{v}_j$ with $\vec{q}$. Each feature vector is then projected with W and $\vec{b}^W$ before computing the weighted average feature vector $\vec{z}$.

$$\vec{z} = \sum_j \left(W\vec{d}_j + \vec{b}^W \right) s_j \tag{5.4}$$

We also tried learning to predict a relevance score directly from concatenated vision and language features, rather than computing the dot product of the features in a latent embedded space. However, the resulting model appeared to learn a salient region weighting scheme that varied little with the language component. The inner-product based relevance was the only formulation we tried that successfully varies with different queries given the same image.

5.2.3 Language Representation

We represent our words with 300-dimensional Google News dataset pre-trained word2vec vectors for their simplicity and compact representation. We are also motivated by the ability of vector-based language representations to encode similar words with similar vectors, which may aid answering open-ended questions. Using means of word2vec vectors, we construct fixed-length vectors for each question-answer pair, which our model then learns to score. In our results section, we show that our vector-averaging language model noticeably outperforms a more complex LSTM-based model from [6],

demonstrating that BOW-like models provide very effective and simple language representations for VQA tasks.

We first tried separately averaging vectors for each word with the question and answer, concatenating them to yield a 600-dimensional vector, but since the word2vec representation is not sparse, averaging several words may muddle the representation. We improve the representation using the Stanford Parser ([86]) to bin the question into additional separate semantic bins. The bins are defined as follows:

- **Bin 1** captures the type of question by averaging the word2vec representation of the first two words. For example, "How many" tends to require a numerical answer, while "Is there" requires a yes or no answer.

- **Bin 2** contains the nominal subject to encode subject of question.

- **Bin 3** contains the average of all other noun words.

- **Bin 4** contains the average of all remaining words, excluding determiners such as "a," "the," and "few."

Each bin then contains a 300-dimensional representation, which are concatenated with a bin for the words in the candidate answer to yield a 1500-dimensional question/answer representation. Figure 5.4 shows examples of binning for the parsed question. This representation separates out important components of a variable-length question while maintaining a fixed-length representation that simplifies the network architecture.

5.2.4 Image Features

The image features from 100 rectangular regions are fed directly into the region-selection layer from a pre-trained network. We first select candidate regions by extracting the top-ranked 99 Edge Boxes ([5]) from the image after performing non-maximum suppression with a 0.2 intersection over union overlap threshold. We found this aggressive thresholding to be important for selecting smaller regions that may be important for some questions, as the

How many birds are in the photo

| How many | birds | photo | are in |

Is there a cat on the car

| Is there | cat | car | on |

What animal is in the picture

| What animal | animal | picture | is in |

Figure 5.4: Example parse-based binning of questions. Each bin is represented with the average of the word2vec vectors of its members. Empty bins are represented with a zero-vector.

top-ranked regions tend to be highly overlapping large regions. Finally, a whole-image region is also added to ensure that the model at least has the spatial support of the full frame if necessary, bringing the total number of candidate regions to 100 per image. While we have not experimented with the number of regions, it is possible that the improved recall from additional regions may improve performance.

We extract features using the VGG-s network ([87]), concatenating the output from the last hidden layer (4096 dimensions) and the pre-softmax layer (1000 dimensions). The pre-softmax classification layer was included to provide a more direct signal for objects from the Imagenet classification task.

5.2.5 Training

Our network architecture is a multi-layer network as seen in Figure 5.3. Our fully connected layers are initialized with Xavier initialization ([88]) and separated with a batch-normalization ([89]) and ReLU layer ([90]). The word2vec text features are fed into the network's input layer, whereas the image region features feed in through the region selection layer.

Our network sizes are set as follows. The 1500 dimensional language features first pass through 3 fully connected layers with output dimensions 2048, 1500, and 1024 respectively. The embedded language features are then passed through the region selection layer to be combined with the vision features. Inside the region selection layer, projections A and B project both vision and language representations down to 900 dimensions before computing their inner product. The exiting feature representation passes through W with an

output dimension of 2048. then finally through two more fully connected layers with output dimensions of 900 and 1 where the output scalar is the question-answer pair score.

The training was especially sensitive to the initialization of the region-selection layer. The magnitude of the projection matrices A, B and W are initialized to 0.001 times the standard normal distribution. We found that low initial values were important to prevent the softmax in selection from spiking too early and to prevent the higher-dimensional vision component from dominating early in the training.

5.3 Experiments

We evaluate the effects of our region-selection layer on the multiple-choice format of the MS COCO Visual Question Answering (VQA) dataset ([6]). This dataset contains 82,783 images for training, 40,504 for validation, and 81,434 for testing. Each image has 3 corresponding questions with recorded free-response answers from 10 annotators. Any response that comes from at least 3 annotators is considered correct. We evaluate on multiple choice task because its evaluation is much less ambiguous than the open-ended response task, though our method could be applied to the latter by treating the most common or likely M responses as a large M-way multiple choice task. We perform detailed baseline comparisons on the validation set and report final scores on the test set.

We evaluate and analyze how much our region-weighting improves accuracy compared to using the whole image or only language (Tables 5.1, 5.2, 5.3) and show examples in Figure 5.8. We also perform a simple evaluation on a subset of images showing that relevant regions tend to have higher than average weights (Figure 5.7). We also show the advantage of our language model over other schemes (Table 5.4).

5.3.1 Comparisons between region, image, and language-only models

We compare our region selection model with several baseline methods, described below. All models use a 10% held-out from train for model selection.

Model	Overall (%)
Word Only	53.98
Word+Whole Image	57.83
Word+Ave. reg.	57.88
Word+Sal. reg.	58.45
Word+Region Sel.	**58.94**
LSTM Q+I ([6])	53.96

Table 5.1: Overall accuracy comparison on Validation. Our region selection model outperforms our own baselines, demonstrating the benefits of selective region weighting.

- **Word-only** We train a network to score each answer purely from the language representation. This provides a baseline to demonstrate improvement due to image features, rather than just good guesses.
- **Word+Whole image** We concatenate CNN features computed over the entire image with the language features and score them using a three-layer neural network, essentially replacing the region-selection layer with features computed over the whole image.
- **Word+Uniform averaged region features** To test that region weighting is important, we also try uniformly averaging features across all regions as the image representation and train as above.
- **Word+Salient region weighting** We include a baseline where each region's weight is computed independently of the language component. We replace the inner product computation between vision and language features with an affine transformation that projects just the vision features down to a scalar, followed by a softmax over all regions. The layer's output is the weighted combination of concatenated vision and language features as before, but using the salient weights.

Table 5.1 shows the comparison of overall accuracy on the validation set, where it is clear our proposed model performs best. The salient weighting baseline alone showed noticeable improvement over the simpler whole image and averaging baselines. We noticed it performed similarly to the whole image baseline on localization dependent categories such as "what color" due to its inability localize on mentioned subjects, but performed similarly to

the proposed model in scene and sport recognition questions due to its ability to highlight discriminative regions. We also include the best-performing LSTM question+image model on val from the authors of ([6]). This model significantly underperforms even our much simpler baselines, which could be partly because the LSTM requires significantly more supervision to match the effectiveness of the word2vec embedding.

We evaluate our model on the test-dev and test-standard partitions in order to compare with additional models from ([6]). In Table 5.2, we include comparisons to the best-performing question+image based models from the VQA dataset paper, as well as a competitive implementation of the whole image+language baseline from [91]. Our model was retrained on train+val data using the same held-out set as before for model selection. Our model significantly outperforms the baselines in the "others" category, which contains the majority of the question types that our model excels at.

Table 5.3 offers a detailed performance summary across various question types, with comparison with word-only, word+whole image, and word+salient baselines. The proposed region selection baseline significantly outperforms the word-only and the attention-free whole-image+word baseline in answering color and scene questions. Surprisingly, the salient attention baseline, which predicts visual attention using only the image, is able to match the performance of the proposed attention method in many cases, most notably in scene questions. However, its color question performance still significantly trails that of the proposed method, suggesting that visual attention for color questions cannot be easily learned without also looking at the question text.

Figure 5.8 shows a qualitative comparison of results, highlighting some of the strengths and remaining problems of our approach. These visualizations are created by soft masking the image with a mask created by summing the weights of each region and normalizing to a max of one. A small blurring filter is applied to remove distracting artifacts that occur from multiple overlapping rectangles. On color questions, localization of the mentioned object tends to be very good, which leads to more accurate answers. On questions such as "How many birds are in the sky?" the system cannot produce the correct answer but does focus on the relevant objects. The third row shows examples of how different questions lead to different focus regions. Notice how the model identifies the room as a bathroom in the third row by focusing on the toilet, and, when confirming that "kite" is the answer to "What is the woman

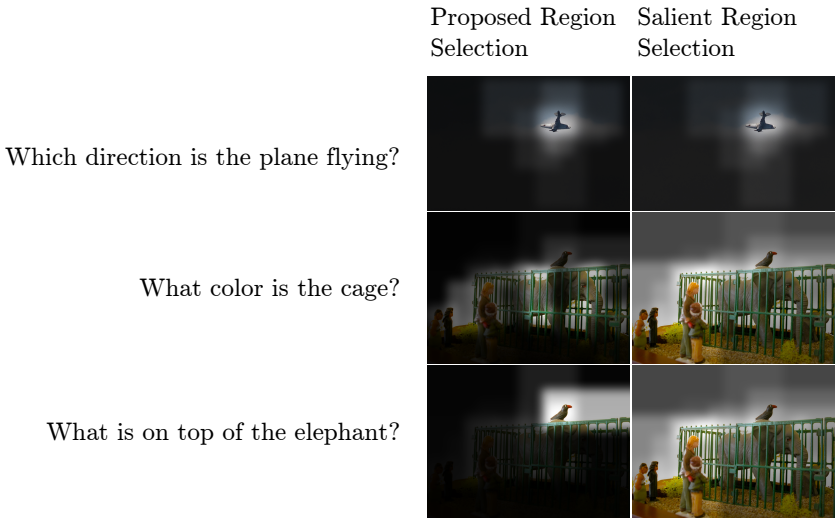

Figure 5.5: Comparison of salient attention, conditioned on only the image, and the proposed attention model that considers both image and query. Many images have predictable saliency, in that it is easy to predict what any question in the image will be about. In the top row of this figure, the salient object is the plane and is predicted by both models with and without considering the query text. In more complex cases such as the bottom two rows, where there are multiple foreground objects, the salient model does a decent job of identifying those over the background, but fails to produce the correct attention map when the query refers to only one of the many possible foreground objects.

flying over the beach?" focuses on the kite, not the woman or the beach. We also compare our proposed attention model with the salient baseline in Figure 5.5. We note that salient attention will behave similarly to our proposed model when the foreground object is easily identifiable and there are few other foreground objects that may be the target of a question.

In Figure 5.6, we show additional qualitative examples of how the region selection varies with question-answer pairs. In the first row, we see the model does more than simply match answer choices to regions. While it does find a matching green region, the corresponding confidence is still low. In addition, we see that irrelevant answer choices tend to have less-focused attention weightings. For example, the kitchen recognition question has most of its weighting on what appears to be a discriminative kitchen patch for the correct choice, whereas the "blue" choice appears to have a more evenly spread out weighting.

Figure 5.6: Comparison of attention regions generated by various question-answer pairings for the same question. Each visualization is labeled with its corresponding answer choice and returned confidence. We show the highlighted regions for the top multiple choice answers and some unrelated ones. Notice that in the first example, while the model clearly identified a green region within the image to match the "green" option, the corresponding confidence was significantly lower than that of the correct options, showing that the model does more than just match answer choices with image regions.

Model	All	Y/N	Num.	Others
test-dev				
LSTM Q+I ([6])	57.17	**78.95**	35.80	43.41
Q+I ([6])	58.97	75.97	34.35	50.33
iBOWIMG ([91])	61.68	76.68	**37.05**	54.44
Word+Region Sel.	**62.44**	77.62	34.28	**55.84**
test-standard				
iBOWIMG ([91])	61.97	76.86	**37.30**	54.60
Word+Region Sel.	**62.43**	**77.18**	33.52	**56.09**

Table 5.2: Accuracy comparison on VQA test sets.

	Word+Reg. Sel.	Word Only	Wrd+Whole	Word+Sal.	Freq.
Overall	**58.9**	54.0	57.8	58.5	100.0
are	**70.2**	56.6	67.5	69.8	7.4
can you	**75.2**	74.0	65.3	74.2	0.4
color	**54.0**	32.6	43.5	46.6	9.8
could	86.7	**90.1**	81.5	88.9	0.3
do/does	75.2	74.6	**75.4**	75.0	3.5
has	73.4	**77.3**	70.8	76.4	0.4
how	28.0	**31.9**	31.9	27.0	1.1
how many	33.0	34.6	**36.6**	34.4	9.0
how many people are	37.1	33.3	**43.1**	36.9	0.9
how many people are in	33.3	23.8	29.2	**34.6**	0.4
is	73.8	75.1	72.0	**76.7**	1.5
is he	75.8	75.6	70.3	**77.9**	0.5
is it	80.8	76.8	77.9	**81.2**	1.7
is that a	75.8	**77.6**	70.5	75.3	0.3
is the	**73.4**	73.1	72.6	73.4	10.2
is there	84.2	84.1	79.7	**86.1**	3.5
is this	76.3	75.6	75.8	**77.4**	7.8
none of the above	56.1	54.6	55.1	**57.9**	4.1
scene: what sport/room	86.2	61.3	76.7	**87.8**	0.9
was	**76.6**	74.7	71.1	75.2	0.4
what	**47.9**	42.6	46.5	47.6	7.5
what animal is	**68.1**	44.1	65.7	65.9	0.4
what are	67.4	52.6	64.8	**68.3**	0.7
what are the	55.8	47.9	**56.2**	54.8	1.5
what brand	45.4	**48.2**	44.0	46.9	0.4
what does the	**38.3**	35.1	35.6	35.8	0.9
what is	**55.4**	47.0	55.3	55.0	13.3
what kind of	**56.4**	45.9	54.3	54.5	2.7
what number is	16.6	17.0	**19.5**	17.3	0.3
what time	**41.5**	38.6	37.7	40.6	0.8
what type of	**55.7**	46.1	52.7	55.5	1.9
where are the	**42.7**	38.2	41.2	42.0	0.6
where is the	41.9	37.4	**42.9**	41.5	1.9
which	45.8	40.4	44.7	**46.2**	1.2
who is	**39.8**	34.7	38.3	37.0	0.5
why	24.5	26.3	24.1	**24.9**	1.1

Table 5.3: Accuracies by type of question on the validation set. Percent accuracy is shown for each subset for our region-based approach, classification using only text, text with a whole-image feature vector, and text with salient attention (attention based only on image). Overall, our region selection scheme outperforms use of whole images by 2% and text-only features by 5%. The learned salient attention model performed surprisingly well. Most notably, it had a similar performance boost over the whole-image baseline on scene questions. The proposed region-selection model still outperforms all baselines on color questions, suggesting that attention for color-identification cannot be easily learned via saliency.

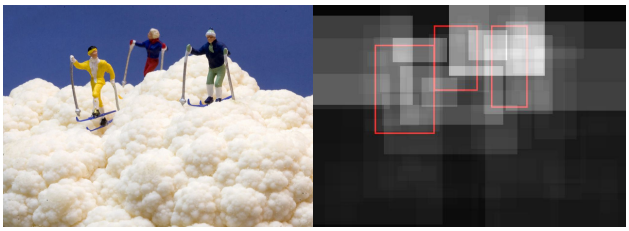

Figure 5.7: Example image with corresponding region weighting. Red boxes correspond to manual annotation of regions relevant to the question: "Are the people real?"

5.3.2 Region Evaluation

Model	Accuracy (%)
Q+A (2-bin)	51.87
parsed(Q)+A (5-bin)	**53.98**

Table 5.4: Language model comparison. The 2-bin model is the concatenation of the question and answer averages. The parsed model uses the Stanford dependency parser to further split the question into 4 bins.

We set up an informal experiment to evaluate the consistency of our region weightings with respect to various types of questions. We manually annotated 205 images from the validation set with bounding boxes considered relevant to answering the corresponding question. An example of the annotation and predicted weights can be seen in Figure 5.7. To evaluate, we compare the average pixel weighting within the annotated boxes with the average across all pixels. Pixel weighting was determined by cumulatively adding each region's selection weight to each of its constituent pixels. We observe that the the mean weighting within the annotated regions was greater than the global average in 148 of the instances (72.2%), often much greater and rarely much smaller. We conduct a more formal evaluation in the next chapter using human annotations provided by Das *et al*[7].

We further investigate the effectiveness of ranking by our region scores in Figure 5.9 by retaining only the top K weighted regions (retained weights are L1 normalized) or only the Kth (1-hot weighting of Kth region). We observe that performance on color-type questions does not improve significantly beyond the first 10 regions, and that performance drops off sharply in the Kth-only experiment. This provides further evidence that our model is able to score relevant regions above the rest.

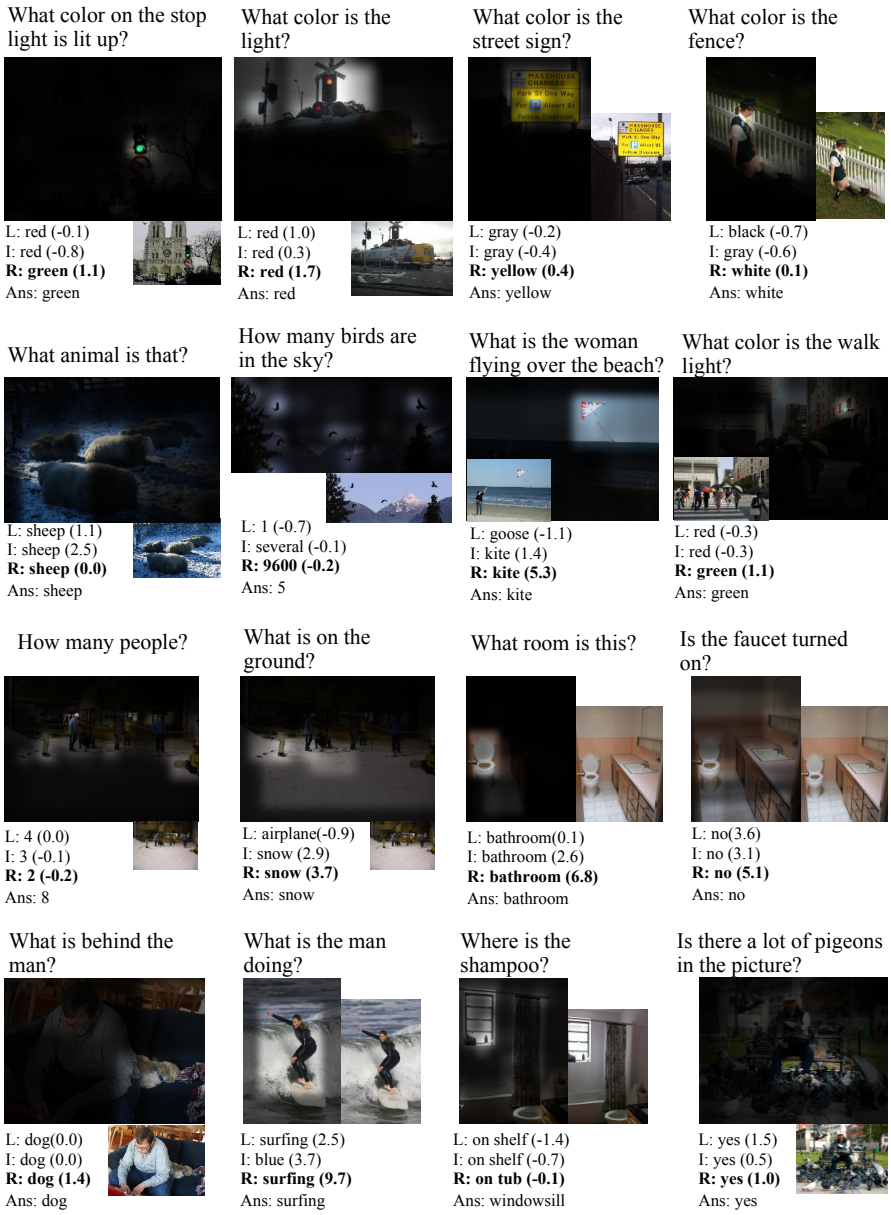

Figure 5.8: Comparison of qualitative results from Val. The larger image shows the selection weights overlaid on the original image (smaller). L: Word only model; I: Word+Whole Image; R: Region Selection. The scores shown are ground truth confidence - top incorrect. Note that the first row shows successful examples in which tight region localization allowed for an accurate color detection. In the third row, we show examples of how weighting varies on the same image due to differing language components.

5.3.3 Language Model

We also compare our parsed and binned language model with a simple two-binned model (one bin averages word2vec of question words; the other aver-

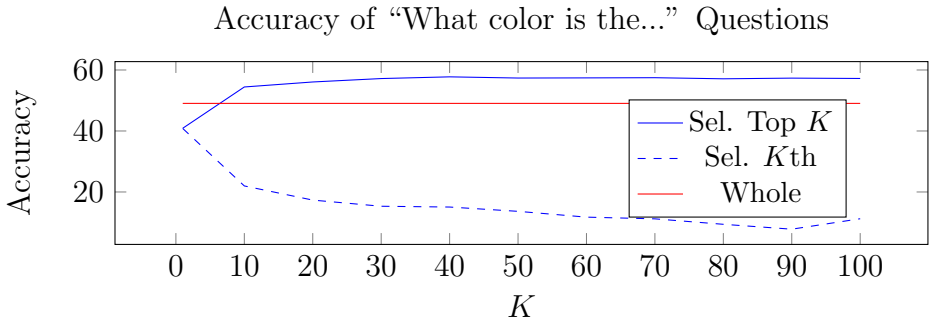

Figure 5.9: Plot of color-based question accuracy with varying number of regions sampled at every 10. The experiment was run on a 10% held-out set on train. We look at using the weighted average of only the top K scoring regions, as well as only the Kth. We include the whole image baseline's accuracy in this category for comparison.

ages answer words) to justify our more complex representation. Each model is trained on the train set and evaluated on the validation set of the VQA real-images subset. The comparison results are shown in Table 5.4 and depict a significant performance improvement using the parsing.

5.4 Conclusion

We presented a model that learns to select regions from the image to solve visual question answering problems. Using internal baselines, we demonstrate that visual attention will improve question-answering accuracy – specifically when it makes sense to exclusively process relevant image regions for the question-answering component. This is apparent in question-types involving color identification, in which focusing on only the relevant parts of the image prevents visual information from irrelevant portions from contributing to the answer. When this type of visual attention is not required, our model is capable of falling back on whole-image attention, allowing it to perform no worse than the simple yet effective whole-image baseline model.

CHAPTER 6

VISUAL ATTENTION FOR VISUAL QUESTION ANSWERING WITH SUPERVISED PHRASE GROUNDING

The previous chapter introduces visual attention for the VQA task in which a latent mapping is learned between various regions of an image and a vectorized representation of a query. This works well in many cases as visual attention for VQA is often a detection task on some of the mentioned nouns or adjectives. Attention for questions such as "What color is the car?," and "Is there a red light?" can be handled with the explicit detection of "car", "red", and "light". While it is interesting that such a mapping can be learned between noisy representations of entire queries and specific regions of an image, directly supervising the word to image region mapping for the straightforward cases should significantly improve the quality of attention maps.

In the following chapter, we look at training an attention-based VQA model that uses direct supervision for mapping individual phrases to parts of the image. While this mapping task is a form of object detection, it differs in that the "labels" are now from the space of natural language (a car can be referred to as "car," "honda," "buggy" etc.) as opposed to being from a fixed, predefined vocabulary. Following the recent work in image captioning, several strongly annotated datasets have been released that ground phrases within a caption to image regions (eg. Flickr30k Entities ([92]) and Visual Genome ([93]). While these datasets focus on providing grounded phrases from statements rather than questions, this distinction does not matter at the phrasal level (eg. "red car" in "Is there a **red car**?" versus "This picture contains a **red car**"). In our work, we incorporate the phrase-level recognition supervision from Genome into the VQA model's training data.

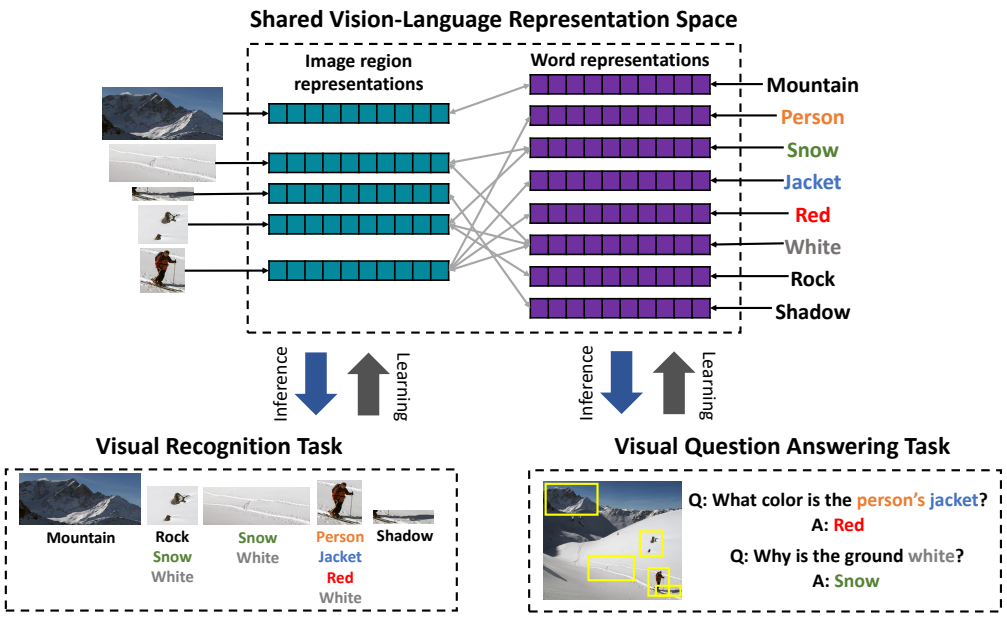

Figure 6.1: **Sharing image-region and word representations across multiple vision-language domains:** The SVLR module projects images and words into a shared representation space. The resulting visual and textual embeddings are then used for tasks like Visual Recognition and VQA. The models for individual tasks are formulated in terms of inner products of region and word representations enforcing an alignment between them in the shared space.

6.1 Method

In order to apply phrase-level recognition supervision to train an attention-based VQA model, we propose a Shared Vision-Language Representation (SVLR) module. The SVLR acts as an intermediate representation that is shared between the phrase recognition task (hereon referred to as visual recognition or VR) and the VQA task, promoting mutually beneficial inductive transfer (see Fig. 6.1) as we will demonstrate later in results.

As with visual attention in the previous chapter, we formulate VR in terms of a joint embedding of textual and visual representations computed by the SVLR module, mapping each region as close to its corresponding phrase in a shared embedding space. For example, the embedding of "dog" should be closer to an embedded region showing a dog than any other object label. We formulate VQA as predicting an answer from a relevant region, where relevance and answer scores are computed from embedded word-region similarities. For example, a region will be considered relevant to "Is the elephant

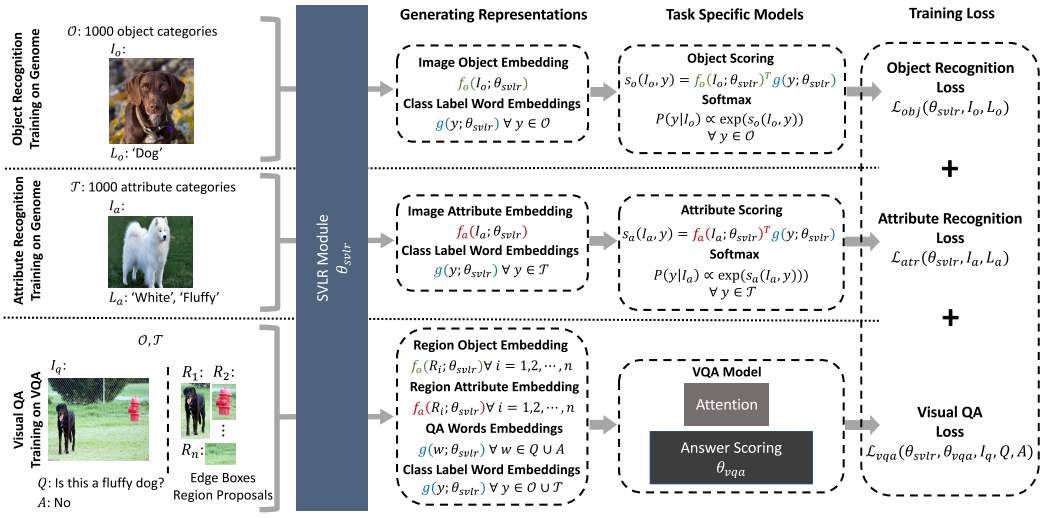

Figure 6.2: **Joint Training on Visual Recognition(VR) and Visual Question Answering(VQA) with the proposed SVLR Module:** The figure depicts sharing of image and word representations through the SVLR module during joint training on object recognition, attribute recognition, and VQA. The recognition tasks use object and attribute labelled regions from Visual Genome while VQA uses images annotated with questions and answers from the VQA dataset. The benefit of joint training is that while the VQA dataset does not provide region groundings of nouns and adjectives in the QA (eg. "fluffy","dog"), this complementary supervision is provided by the Genome recognition dataset. Models for each task involve image and word embeddings produced by SVLR module or their inner products (See Fig 6.3 for VQA model architecture).

wearing a pink blanket?" if the embedded "pink" and either "elephant" or "blanket" are close to the embedded region. Similarly, the answer score considers embedded similarities, but in a more comprehensive manner. We emphasize that the same word-region embedding is learned for both VR and VQA. Our experiments show that formulating both tasks in terms of the SVLR module leads to better cross-task transfer than if features are shared through multitask learning but without exploiting the alignment between words and regions.

6.1.1 SVLR

The SVLR module converts words and image-regions into feature representations that are aligned to each other and shared across tasks. As shown in

Fig. 6.2, the word and region representations required for object recognition, attribute recognition, and VQA are computed through the SVLR module. By specifically formulating each task in terms of inner products of word and region representations and training on all tasks jointly, we ensure each task provides a consistent, non-conflicting training signal for aligning words and region representations. During training, the joint-task model is fed batches containing training examples from each tasks' dataset.

Word Representations: The representation $g(w)$ for a word w is constructed by applying two fully connected layers (with 300 output units each) to pretrained word2vec representation [83] of w and ReLU after the first layer.

Region Representations: A region R is represented using two 300 dimensional feature vectors $f_o(R)$ and $f_a(R)$ that separately encode the objects and attributes contained. We used two representations instead of one to encourage disentangling of these two factors of variation. For example, we do not expect "red" to be similar to "apple", but we expect $f_o(R)$ and $f_a(R)$ to be similar to $g(\text{“}red\text{”})$ and $g(\text{“}apple\text{”})$ if R depicts a red apple. The features are constructed by extracting the average pooled features from Resnet [37] pretrained on ImageNet and then passing through separate object and an attribute networks. Both networks consist of two fully connected layers (with 2048 and 300 output units) with batch normalization [89] and ReLU activations.

6.1.2 Visual Recognition using SVLR

Inference

The visual recognition task is to classify image regions into one or more object and attribute categories. The classification score for region R and object category w is $f_o^T(R)g(w)$. The classification score for an attribute category v is $f_a^T(R)g(v)$. Attributes may include adjectives and adverbs (e.g., "standing"). Though our recognition dataset has a limited set of object categories $\mathcal{O}$ and attribute categories $\mathcal{T}$, our model can produce classification scores for any object or attribute label given its word2vec representation. In

experiments, the $\mathcal{O}$ and $\mathcal{T}$ consist of 1000 most frequent object and attribute categories in the Visual Genome dataset [93]. We do not use ImageNet class labels as they do not provide attribute classes and the language statistics of Genome should be closer to those in VQA questions.

Training

Our VR model is trained using the Visual Genome dataset which provides image regions annotated with object and attribute labels. VR uses only the parameters for the embedding functions f_o, f_a and g that are part of the SVLR module. The parameters of f_o receive gradients from the object loss while those of f_a receive gradients from the attribute loss. The parameters of word embedding model g receive gradients from both losses.

Object loss: We use a multi-label classification loss, since object classes may not be mutually exclusive due to hypernyms (e.g., "man" *is a* "person") and synonyms. For a region R_j, we denote the set of annotated object categories and their hypernyms extracted from WordNet [94] by $\mathcal{H}_j$. The object loss ensures that the true labels and their hypernyms score more than all other object labels by a margin η_{obj}. For a batch with M samples $\{(R_j, \mathcal{H}j)\}_{j=1}^{M}$ the object loss is:

$$\mathcal{L}_{obj} = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{|\mathcal{H}j|}\sum_{l\in\mathcal{H}j}\frac{1}{|\mathcal{O}|}\sum_{k\in\mathcal{O}\setminus\mathcal{H}j} \max\{0, \eta_{obj} + f_o^T(R_j)g(k) - f_o^T(R_j)g(l)\} \quad (6.1)$$

Attribute Loss: The attribute loss is a multi-label classification loss with two differences from object classification. Unlike in object classification, labels are rarely mutually exclusive. We account for this by using the independent cross entropy losses for each attribute. We also weight the samples based on fraction of positive labels in the batch to balance the positive and negative labels in the dataset. For a batch with M samples $\{(R_j, \mathcal{T}_j)\}_{j=1}^{M}$ where $\mathcal{T}_j$ is the set of attributes annotated for region R_j, the attribute loss

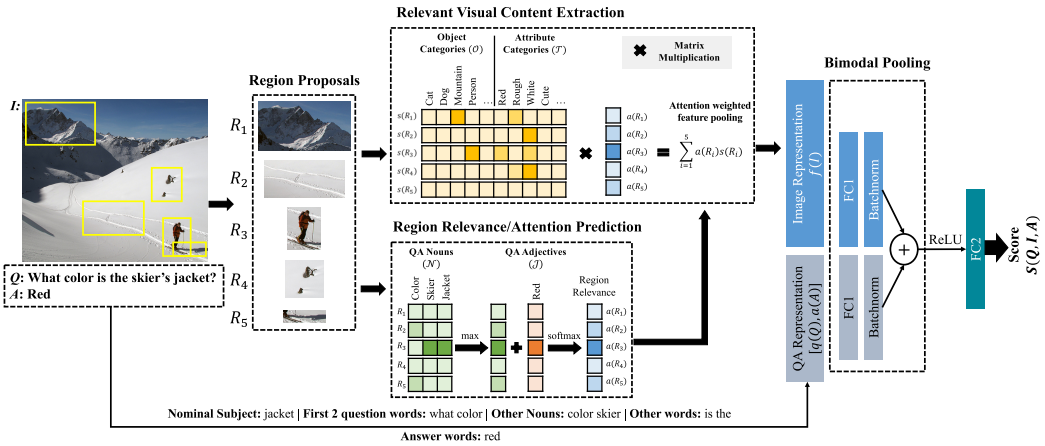

Figure 6.3: **Inference in our VQA model:** The image is first broken down into Edge Box region proposals[5]. Each region R is represented by visual category scores $s(R) = [s_o(R), s_a(R)]$ obtained using the visual recognition model. Using the SVLR module, the regions are also assigned an attention score using the inner products of region features with representations of nouns and adjectives in the question and answer. The region features are then pooled using the relevance scores as weights to construct the *attended* image representation. Finally, the image and question/answer representations are combined and passed through a neural network to produce a score for the input question-image-answer triplet.

is:

$$\mathcal{L}_{atr} = \frac{1}{M}\sum_{j=1}^{M}\sum_{t\in\mathcal{T}} \mathbb{1}\left[t \in \mathcal{T}_j\right](1-\Gamma(t))\log\left[\sigma(f_a^T(R_j)g(t))\right] + \mathbb{1}\left[t \notin \mathcal{T}_j\right]\Gamma(t)\log\left[1-\sigma(f_a^T(R_j)g(t))\right] \quad (6.2)$$

where σ is a sigmoid activation function and $\Gamma(t)$ is the fraction of positive samples for attribute t in the batch.

6.1.3 Visual Question Answering using SVLR

Our VQA model is illustrated in Fig. 6.3. The input to our VQA model is an image, a question and multiple answer choices. Regions are extracted from the image using Edge Boxes [5]. The same SVLR module used by VR (Sec. 6.1.2) is explicitly applied to VQA for attention and answer scoring.

Our system assigns attention scores to each region according to how well it matches words in the question/answer, then scores each answer based on the question, answer, and attention-weighted scores for all objects ($\mathcal{O}$) and attributes ($\mathcal{T}$).

Attention Scoring: Unlike other attention models [95, 96] that are free to learn any correlation between regions and question/answers, our attention model encodes an explicit notion of vision-language grounding. Let $\mathcal{R}$ be the set of region proposals extracted from the image, and $\mathcal{N}$ and $\mathcal{J}$ denote the set of nouns and adjectives in the (Q, A) pair. Each region $R \in \mathcal{R}(I)$ is assigned an attention score $a(R)$ as follows:

$$a'(R) = \max_{n \in \mathcal{N}} f_o^T(R) g(n) + \max_{j \in \mathcal{J}} f_a^T(R) g(j) \tag{6.3}$$

$$a(R) = \frac{\exp(a'(R))}{\sum_{R' \in \mathcal{R}(I)} \exp(a'(R'))} \tag{6.4}$$

Thus, a region's attention score is the sum of maximum adjective and noun scores for words mentioned in the question or answer (which need not be in sets $\mathcal{O}$ and $\mathcal{T}$).

Image Representation: To score an answer, the content of region R is encoded using the VR scores for all objects and attributes in $\mathcal{O}$ and $\mathcal{T}$, as presence of unmentioned objects or attributes may help answer the question. The image representation is an attention-weighted average of these scores across all regions:

$$f(I) = \sum_{R \in \mathcal{R}(I)} a(R) \begin{bmatrix} s_o(R) \\ s_a(R) \end{bmatrix} - \tag{6.5}$$

where I is the image, $s_o(R)$ are the scores for 1000 objects in $\mathcal{O}$ for each image region R, $s_a(R)$ are the scores for 1000 attributes in $\mathcal{T}$, and $a(R)$ is the attention score.

Question/Answer Representation: To construct representations $q(Q)$ and $a(A)$ for the question and answer, we follow Shih et al. [97] (see 5.2.3) , dividing question words into 4 bins, averaging word representations in each bin, and concatenating the bin representations resulting in a 1200 ($= 300 \times 4$) dimensional vector $q(Q)$. The answer representation $a(A) \in \mathbb{R}^{300}$ is obtained by averaging the word representations of all answer words. The word representations used here are produced by the SVLR module.

Answer Scoring: We combine the image and Q/A representations to jointly score the (Q, I, A) triplet.

To ensure equal contribution of language and visual features, we perform batch normalization [89] on linear transformations of these features before adding them together to get a bimodal representation $\beta(Q, I, A) \in \mathbb{R}^{2500}$. Specifically,

$$\beta(Q, I, A) = \mathcal{B}_1(W_1 f(I)) + \mathcal{B}_2\left(W_2 \begin{bmatrix} q(Q) \\ a(A) \end{bmatrix}\right) \quad (6.6)$$

where $\mathcal{B}_1, \mathcal{B}_2$ denote batch normalization layers and $W_1 \in \mathbb{R}^{2500\times 2000}$ and $W_2 \in \mathbb{R}^{2500\times 1500}$ define the linear transformations. The bimodal representation is then computed as:

$$\mathcal{S}(Q, I, A) = W_3 \,\mathrm{ReLU}(\beta(Q, I, A))$$

with $W_3 \in \mathbb{R}^{1\times 2500}$.

Training: We use the VQA dataset [6] for training parameters of our VQA model: W_1, W_2, W_3, and scales and offsets of batch normalization layers. In addition, the VQA loss backpropagates into f_o, f_a which are part of the SVLR module. Each sample in the dataset consists of a question Q about an image I with list of answer options including a positive answer A^+ and N negative answers $\{A^-(i)|i = 1, \cdots, N\}$.

The VQA loss encourages the correct answer A^+ to be scored higher than all incorrect answer options $\{A^-(i)|i = 1, \cdots, N\}$ by a margin η_{ans}. Given batch samples $\{(Q_j, I_j, A_j)\}_{j=1}^P$, the loss is written as

$$\mathcal{L}_{ans} = \frac{1}{NP}\sum_{j=1}^{P}\sum_{i=1}^{N} \max\{0, \eta_{ans} + \mathcal{S}(Q_j, I_j, A_j^-(i)) - \mathcal{S}(Q_j, I_j, A_j^+)\} \quad (6.7)$$

Figure 6.4: **Interpretable inference in VQA:** Our model produces interpretable intermediate computation for region relevance and object/attribute predictions for the most relevant regions. Our region relevance explicitly grounds nouns and adjectives from the Q/A input in the image. In addition to attention, we show object and attribute predictions for the most relevant region identified for a few correctly answered questions. The relevant regions are visualized by applying a mask generated from relevance scores projected back to their source pixel locations.

Accuracies on VQA Val	**what color**	**what is the (wo)man/person**	**what is in/on**	**what kind/ type/animal**	**what room/sport**	can/could/ does/do/has	what does/ number/name	what brand	which/who	what is/are	why/how	how many	what time	where	is/are/was	none of the above	other	number	yes/no	overall accuracy
VQA Only	53.5	70.5	53.6	56.8	89.8	81.8	41.9	45.9	49.0	58.3	**33.8**	38.4	**53.9**	45.8	80.2	56.0	54.5	39.2	82.1	62.9
Joint Multitask	59.4	71.8	54.6	58.3	91.0	81.9	**43.8**	46.4	50.8	59.2	32.3	**39.4**	**53.9**	47.0	80.4	57.1	56.7	**39.8**	82.2	64.1
Joint SVLR	**62.1**	**74.1**	**57.9**	**60.0**	**91.1**	**82.8**	41.6	**52.9**	**52.0**	**61.1**	33.6	39.0	51.3	**48.6**	**81.4**	**58.5**	**58.8**	38.8	**83.0**	**65.3**

Table 6.1: **Inductive transfer from VR to VQA through SVLR in joint training:** We evaluate the performance of our model with the SVLR module trained jointly with VR and VQA supervision (provided by Genome and VQA datasets respectively) on the validation set of the multiple-choice VQA task. We compare this jointly-trained model to a model trained on *only* VQA data. We also compare to a traditional multitask learning setup that is jointly trained on VQA and VR and shares visual features but *does not* use the object and attribute word embeddings for recognition. While multitask learning outperforms the VQA-only model, using the SVLR module doubles the improvement. Our model is most suited for the question types in bold that require visual recognition without specialized skills like counting or reading. In this setting we train on Genome VR data and apply to VQA val. Details in Sec 6.2.2.

6.2 Experiments

Our experiments investigate the extent to which using SVLR (Shared Vision-Language Representation) as a core representation improves inductive transfer in multitask learning. We first analyze how including the VR (recognition) task improves VQA (Sec. 6.2.2, Tab. 6.1). We find that using SVLR doubles the improvement compared to standard multitask learning. We then analyze improvement to VR due to training with (weakly supervised) VQA (Sec. 6.2.3, Fig. 6.5). We find moderate overall improvements (1.2%), with the largest improvements for classes that have few VR training examples. For reference, we include results of our VQA system trained with ResNet-152 architecture on val, test-dev, test-std, along with state-of-the-art (Tab. 6.2). Finally, we compare the attention-map quality with the latent attention learned in the previous chapter, using the annotated human attention benchmark from Das *et al* [7].

6.2.1 Datasets

Our model is trained on two separate datasets, one for VQA supervision, the other for visual recognition (attributes and object classification). We

use the image-question-answer annotation triplets from Antol et al. [6] and bounding box annotations for object and attribute categories from Visual Genome [93]. The train-val-test splits for the datasets are as follows.

VQA: We split the *train* set into *train-subset* and *train-held-out* and use the latter for model selection. The *train-subset* consists of 236,277 (Q, I, A) samples whereas *train-held-out* contains 12,072 samples. The *val* and *test* set contain 121,512 and 244,302 samples respectively. There are exactly 3 questions per image. We use the VQA validation set for analyzing performance for specific question types.

Visual Genome: We use only images from Visual Genome not in VQA (overlap identified using md5 hash comparisons). The selected images were divided into *train-val-test* using an 85-5-10 split, yielding 1,565,280, 90,212 and 181,141 annotated regions in each. We use *val* for selecting the model for evaluating recognition performance.

6.2.2 Inductive Transfer from VR to VQA

In Table 6.1, we analyze the role of SVLR module for inductive transfer in joint training.

Joint Training: During joint training, the VR models and VQA model are simultaneously trained using object and attribute annotations from Genome, and Q/A annotations from the VQA dataset. The common approach to joint training in vision research is to use a common network for extracting image features (e.g. class logits from ResNet), which feeds into the task-specific networks as input. We refer to this approach in Table 6.1 as *Joint Multitask*. This baseline is implemented by replacing $g(y)$ (see Fig. 6.2), with a fixed set of vectors h_y for each of the predetermined 1000 object and 1000 attribute categories in the VR models. The embedding $g(y)$ is still in the VQA model, but is no longer shared across tasks. Our proposed *Joint SVLR* outperforms VQA-only by 2.4%, doubling the 1.2% improvement achieved by *Joint Multitask*. Our formulation of VR and VQA tasks in terms of shared word-region representations more effectively transfers recognition knowledge from VR than shared features. On questions that typically involve recognition (in bold in Table 6.1), the gain is typically larger. For example, *what color* questions improve by 8.6% due to SVLR.

Surprisingly, pre-training the visual classifiers on Genome prior to joint training performs worse than the model trained jointly from scratch: 63.7% compared to 65.3%.

6.2.3 Inductive Transfer from VQA to VR

To analyse inductive transfer from VQA to VR tasks we compare the performance of our SVLR based model trained jointly on VQA and VR data with a model trained only on Genome data. Genome *test* set is used for evaluation. We observe an increase in the overall object recognition accuracy from 43.3% to 44.5%, whereas average attribute accuracy remained unchanged at 36.9%. In Fig. 6.5, we show that nouns that are relatively rare in Genome (left columns) but have 20 or more examples in VQA (upper rows) benefit most from weak supervision provided by VQA. On average, we measure improvement from 21% to 32% for the 8 classes that have fewer than 125 examples in Genome train but occur more than 160 times in VQA questions. We conducted the same analysis on Genome attributes, but did not observe any notable pattern, possibly due to the inherent difficult in evaluating the multi-label attribute classification problem (the absence of attributes is not annotated in Genome).

6.2.4 Interpretable Inference for VQA

As shown in Fig. 6.4, our VQA model produces interpretable intermediate outputs such as region relevance and visual category predictions, similar to [98]. The choice of answer in each case is easily explained by the object and attribute predictions associated with the most relevant regions identified by the model. Because relevance is posed as explicit localization of question and answer nouns and adjectives, it is possible to qualitatively evaluate the quality of relevance prediction by verifying that the predicted relevant regions match the said words. This also provides greater insight into the failure modes as shown in Fig. 6.6.

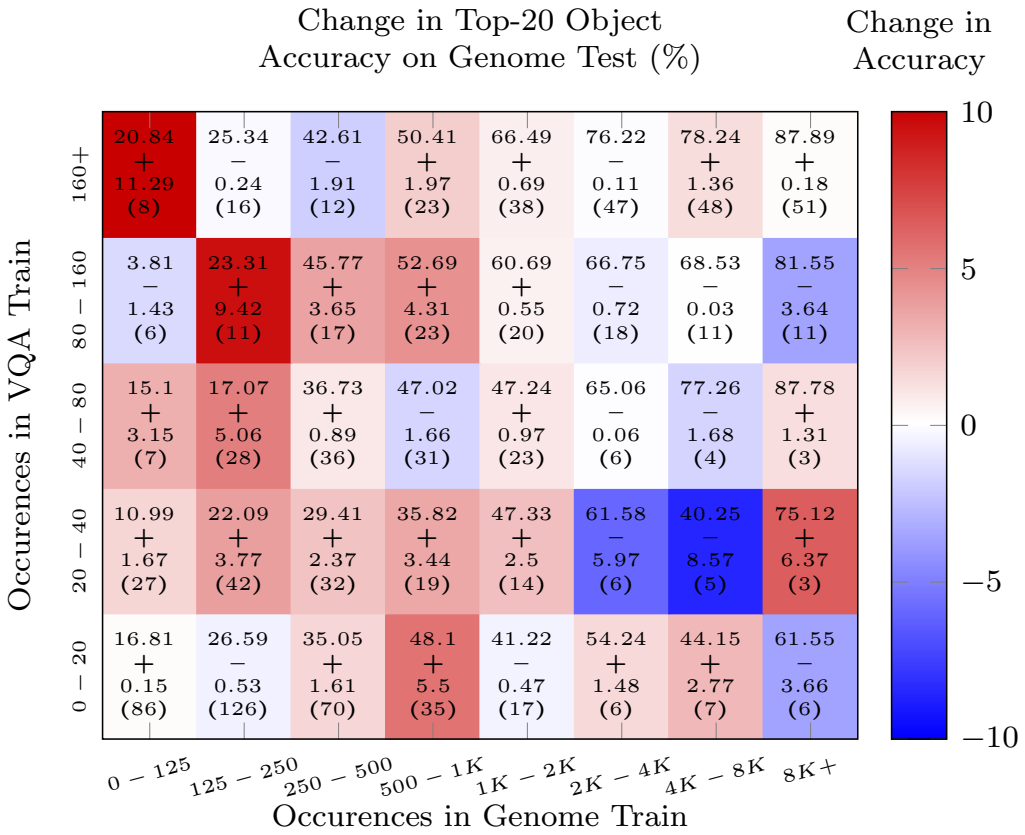

Figure 6.5: **Inductive Transfer from VQA to Object Recognition:** Each cell's color reflects the average accuracy improvement for classes within the corresponding frequency ranges of both datasets from training on Genome-only to training on Genome and VQA. Most gains are in rare Genome nouns with higher frequency in the VQA dataset (top left corner), suggesting that the weak supervision provided by training VQA attention helped to augment performance via the SVLR. The numbers in each cell show the Genome-only mean accuracy +/- the change due to SVLR multitask training, followed by the number of classes in the cell in parentheses.

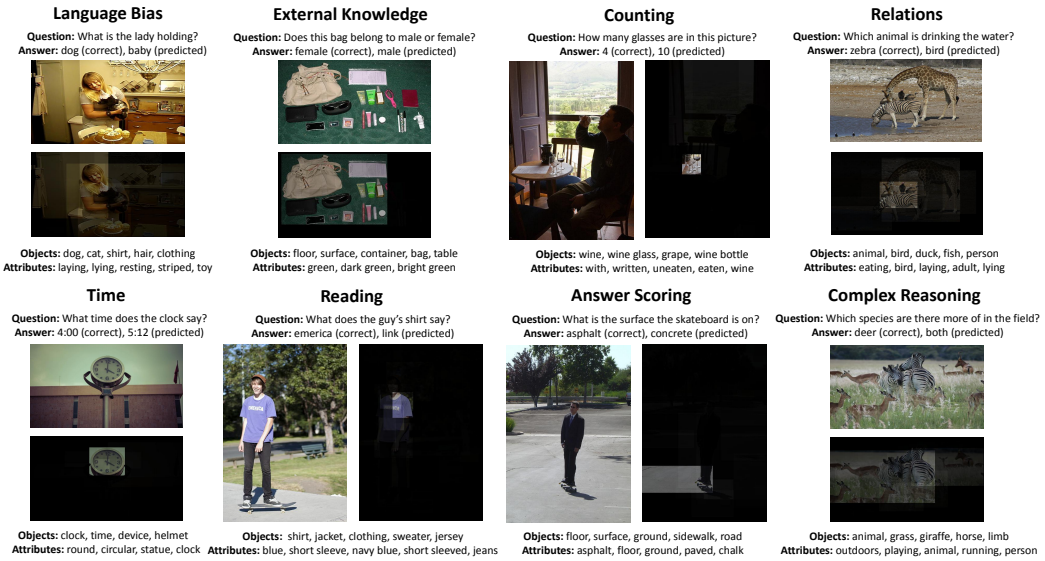

Figure 6.6: **Failure modes:** Our model cannot count or read, though it will still identify the relevant regions. It is blind to relations and thus fails to recognize that *birds*, while present in the image, are not *drinking water*. The model may give a low score to the correct answer despite accurate visual recognition. For instance, the model observes *asphalt* but predicts *concrete*, likely due to language bias. A clear example of an error due to language bias is in the top-left image as it believes the lady is holding a *baby* rather than a *dog*, even though visual recognition confirms evidence for dog. Finally, our model fails to answer questions that require complex reasoning involving comparison of multiple regions.

6.2.5 Comparison with Human Attention

We now look at how the learned visual attention correlates with that of humans. We evaluate our models' attention maps on a collection of human attention maps from Das *et al* [7]. This collection of attention maps provides 3 human attention annotations per question for 1374 question-image pairs in the validation set of VQA. Annotations were collected by posing the question-image pairs to annotators, but with the image portion blurred out. The annotators were allowed to sharpen parts of the image to answer the question, and the parts they chose to sharpen were recorded as the human attention.

We compute correlation following the method used in Das *et al.* We first resize all attention maps to 14 by 14 using MATLAB's imresize function with bilinear interpolation before comparing and add random noise of the order 10^{-14} to the human attention maps to account for equally-weighted regions.

	Where To Look [97]	FDA [99]	MLP [3, 4]	MCB [100]	Co-Attention [96]	Ours
val	58.9	-	63.6	-	-	66.2
test-dev	62.4	64.0	65.9	69.9	65.8	64.8
test-std	63.5	64.2	-	-	66.1	64.8
Trained on	*train+val*	-	*train*	*train+val*	*train+val*	*train*

Table 6.2: **VQA performance on val and test sets:** Because these systems vary in many ways, our internal comparisons are more instructive, but we include these for reference. For test accuracy, it is unclear whether FDA uses *val* in training. The MLP results were obtained using the implementation provided by [3]. The original MLP implementation [4] using Resnet-101 yields 64.9 and 65.2 on *test-dev* and *test-std* respectively. MCB reports only *test-dev* accuracy for the directly comparable model (final without ensemble).

We compute the Spearman rank correlation coefficient between the human attention maps and that of our models, averaged across all 3 annotations per question. Das *et al* further suggested that the relative correlation performance between models is more apparent when comparing only on attention maps that cannot be handled by simply biasing attention towards the center of the image. To incorporate this, we use the synthetic center-focused heat map (see fig. 6.7) provided by the authors. We then compute the Spearman rank correlation between this mask and all human attention maps and threshold at various levels. Each threshold defines a subset for which the correlation between the center-focused heat map and the human attention is less than or equal to the current threshold value.

We use three models in this comparison: the proposed model from the previous chapter (WTL), the salient baseline from the previous chapter (attention conditioned on only image), and the SLVR-trained attention model from this chapter. The comparison results at various threshtolds is seen in Fig. 6.9. Notice that as the threshold approaches 1.0, the center baseline (using the center-focused heat map as the model attention) actually outperforms all learned models, suggesting that the question-relevant region is most often located in the center of the image. The learned attention models start to outperform the baselines as the threshold decreases, with the SVLR attention performing significantly better than both the WTL and the salient baseline. Some comparable qualitative results are shown in Fig. 6.8.

Figure 6.7: Synthetic center-focused image baseline provided by the authors of Das *et al* [7]. This image was used to represent a baseline attention model that always focuses on the center of the image. By computing the correlation between the human attention maps and this one, we are able to identify low correlation subsets of the dataset in which the human subjects looked away from the image center.

6.3 Conclusion

We introduced a VQA model in which the visual attention is formulated in a much more explicit manner than in the previous chapter. We isolate visual phrases in the queries such as mentioned nouns and adjectives and use additional training data from Visual Genome to directly learn their correspondence with image regions. We further demonstrated that in the two tasks that we train for (visual recognition and question answering), joint training using our SVLR formulation allows for inductive transfer, mutually benefiting both tasks. Finally, we compare its generated attention maps with that of the model from the previous chapter and demonstrate a significantly stronger correlation with human visual attention from annotations provided by Das *et al* [7]. While overall performance in VQA may not have improved significantly, we argue that performance on questions requiring explicit attention (e.g. color questions) have (up to 62% accuracy) when compared to both the attention model in the previous chapter as well as baselines within this chapter.

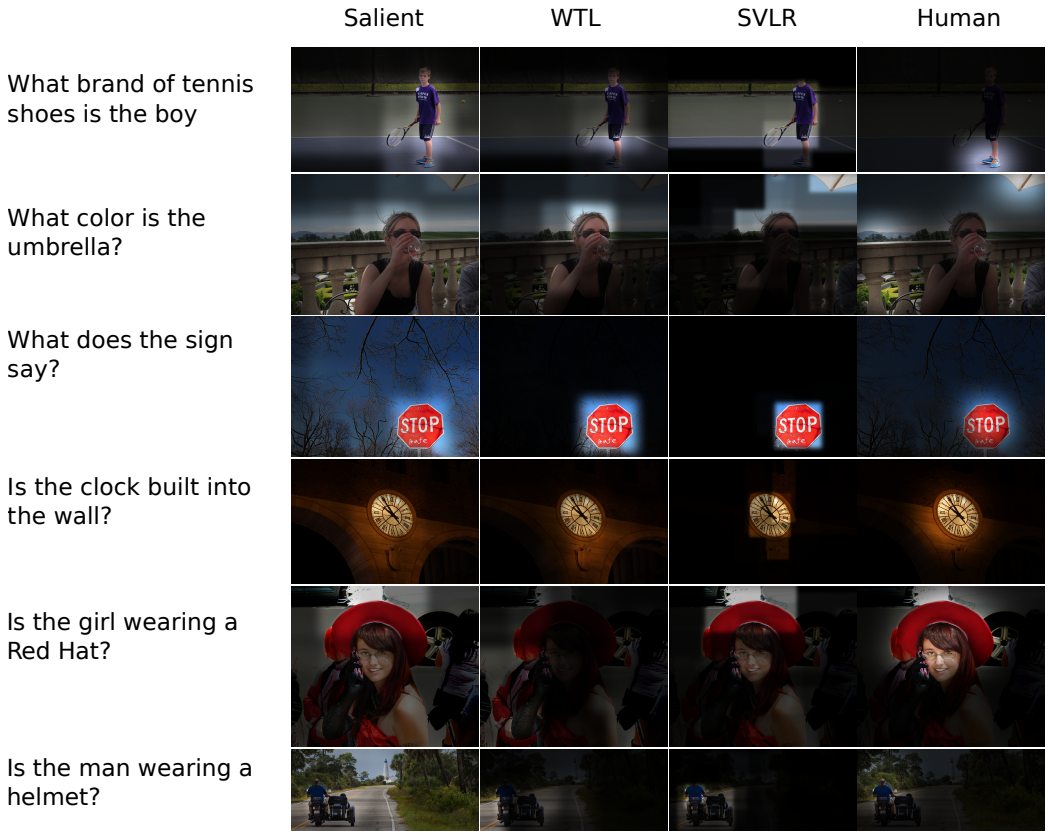

Figure 6.8: Qualitative comparison of attention maps from various models. Saliency generally corresponds pretty well with what questions ask about. Compared to the WTL model, the SVLR model's attention is typically much more focused. Regions deemed irrelevant by the SVLR seem to be more readily downweighted than in the WTL and Salient cases. Note that Gaussian smoothing was used on the attention masks for Salient, WTL, and SVLR for visualization purposes only.

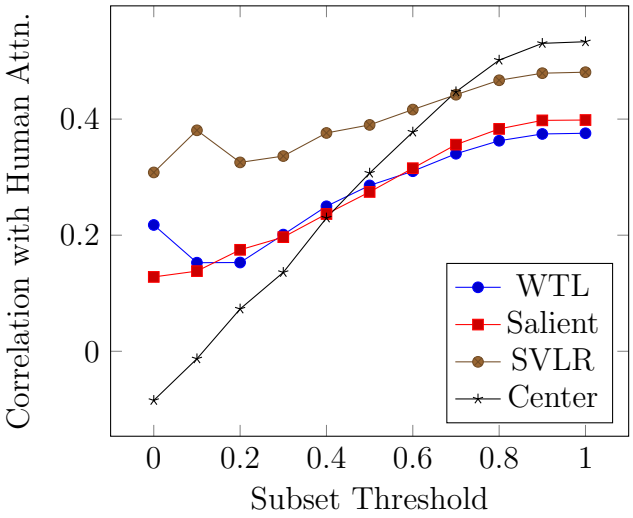

Figure 6.9: Mean Spearman rank-correlation coefficients between model attention and human attention at various threshold. The threshold points define subsets of the dataset for which the human attention correlation with the synthetic center heatmap is below the current threshold value. For example: the first sample point of each curve is the mean correlation of each model with human attention, measured on a subset in which the human attention's correlation with the center heatmap is less than or equal to 0. WTL and Salient are the proposed model and salient attention baseline from the previous chapter. The Center baseline is the correlation of the center heatmap measured against all examples in the current subset. As can be seen, the attention of the proposed SVLR model significantly outperforms those of the models from the previous chapter at all threshold levels. The WTL slightly outperforms its corresponding strong salient baseline up to the threshold at 0.6. As the threshold approaches 1, the synthetic center heatmap baseline outperforms all proposed models, confirming that the majority of the questions are asking about something in the center of the image. Note that there were only 11 examples in ≤ 0 threshold and 748 in the ≤ 0.6 threshold.

CHAPTER 7

CONCLUSIONS

We have looked at various ways of tackling several computer vision tasks by incorporating visual attention. By learning to identify task-relevant image regions from a pool of object proposals, we were able to achieve remarkable results in keypoint localization, fine-grained image recognition, and visual question answering.

In keypoint localization, we found that learning to score sampled regions by keypoint visibility allowed us to make multiple independent keypoint predictions at various locations, scales, and aspect ratios. Importantly, each independent prediction is also confident that the keypoint-to-localize was at least somewhere within the sampled area. Extending the keypoint localization results to conduct part-aligned fine-grained classification led to improved classification performance compared to previous works.

Our work also established initial benchmarks for applying visual attention in high-level vision language tasks such as VQA. Using the flexible deep learning frameworks available, we were able to setup and train complex models capable of adapting their behavior with respect to natural language input. Our work demonstrated how to train visual attention for text-based question-answering in an end-to-end fashion using only QA supervision. In order answer questions more accurately, the model learned to identify task-relevant image regions as a latent task.

We also looked at introducing more direct supervision to the attention task by leveraging datasets with phrase-grounding annotations to simultaneously train the model for image to phrase matching. This was achieved by explicitly formulating the attention task within the VQA model as the mapping of noun and adjective phrases to relevant parts of the image, thereby allowing the VQA model to directly benefit from better phrase-to-image matching accuracy.

Our work addresses the important question of whether visual attention

matters at all for tasks such as VQA. Comparing to internal attention-free baselines, we demonstrated that attention helped the most in localization-dependent tasks such as color or texture identification, in which using any other part of the image but the target in question would have introduced only noise to the inference procedure. Further, the behavior of our learned attention models positively correlates with that of humans on the VQA task, suggesting that where the models choose to look is reasonable and should be interpretable by humans.

Images provide a lot of information, much of which is irrelevant to the task at hand. Furthermore, as we scale our models to target real-world applications, the amount of available data increases for both training and inference. As such, visual attention will continue to play a role in developing accurate and computationally efficient models.

CHAPTER 8

REFERENCES

[1] J. Liu and P. N. Belhumeur, "Bird part localization using exemplar-based models with enforced pose and subcategory consistency," in *ICCV*. IEEE, 2013, pp. 2520–2527.

[2] J. Liu, Y. Li, and P. N. Belhumeur, "Part-pair representation for part localization," in *ECCV*. Springer, 2014, pp. 456–471.

[3] A. Mallya, "simple-vqa: Code implementing VQA MLP baseline from Revisiting Visual Question Answering Baselines," https://github.com/arunmallya/simple-vqa, 2016, [Online; accessed 14-Nov-2016].

[4] A. Jabri, A. Joulin, and L. van der Maaten, "Revisiting visual question answering baselines," in *ECCV*. Springer, 2016, pp. 727–739.

[5] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*. Springer, 2014, pp. 391–405.

[6] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *ICCV*, 2015.

[7] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra, "Human attention in visual question answering: Do humans and deep networks look at the same regions?" in *EMNLP*, 2016.

[8] A. Yarbus, "Eye movements during perception of complex objects," *Eye Movements and Vision*, pp. 171–196, 1967.

[9] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[10] J. M. Wolfe, K. R. Cave, and S. L. Franzel, "Guided search: an alternative to the feature integration model for visual search." *Journal of Experimental Psychology: Human perception and performance*, vol. 15, no. 3, p. 419, 1989.

[11] J. M. Wolfe, "Guided search 2.0 a revised model of visual search," *Psychonomic bulletin & review*, vol. 1, no. 2, pp. 202–238, 1994.

[12] S. Frintrop, E. Rome, and H. I. Christensen, "Computational visual attention systems and their cognitive foundations: A survey," *ACM Transactions on Applied Perception*, vol. 7, no. 1, pp. 6:1–6:39, Jan. 2010. [Online]. Available: http://doi.acm.org/10.1145/1658349.1658355

[13] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *CVPR*. IEEE, 2010, pp. 73–80.

[14] I. Endres and D. Hoiem, "Category-independent object proposals with diverse ranking," *PAMI*, vol. 36, no. 2, pp. 222–234, 2014.

[15] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[16] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *CVPR*. IEEE, 2010, pp. 3241–3248.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[18] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 8, pp. 1558–1570, 2015.

[19] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779–788.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *arXiv preprint arXiv:1512.02325*, 2015.

[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.

[23] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[24] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *CVPR*, 2013, pp. 923–930.

[25] I. Endres, K. J. Shih, J. Jiaa, and D. Hoiem, "Learning collections of part models for object recognition," in *CVPR*, 2013, pp. 939–946.

[26] K. Shih, I. Endres, and D. Hoiem, "Learning discriminative collections of part detectors for object recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 1, pp. 1–1.

[27] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *ICCV*. IEEE, 2011, pp. 89–96.

[28] B. Hariharan, J. Malik, and D. Ramanan, "Discriminative decorrelation for clustering and classification," *ECCV*, pp. 459–472, 2012.

[29] S. Singh, A. Gupta, and A. Efros, "Unsupervised discovery of mid-level discriminative patches," *ECCV*, pp. 73–86, 2012.

[30] C. Doersch, A. Gupta, and A. A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Advances in Neural Information Processing Systems*, 2013, pp. 494–502.

[31] J. Sun and J. Ponce, "Learning discriminative part detectors for image classification and cosegmentation," in *CVPR*, 2013, pp. 3400–3407.

[32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[38] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *NIPS*, 2014, pp. 487–495.

[39] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*. IEEE, 2014, pp. 580–587.

[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, June 2016.

[41] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431–3440.

[42] P. O. Pinheiro, R. Collobert, and P. Dollar, "Learning to segment object candidates," in *Advances in Neural Information Processing Systems*, 2015, pp. 1990–1998.

[43] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, pp. 818–833.

[44] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[45] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *arXiv preprint arXiv:1502.03044*, 2015.

[46] S. Sukhbaatar, J. Weston, R. Fergus et al., "End-to-end memory networks," in *NIPS*, 2015, pp. 2440–2448.

[47] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on computers*, vol. 100, no. 1, pp. 67–92, 1973.

[48] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International journal of computer vision*, vol. 61, no. 1, pp. 55–79, 2005.

[49] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation," in *BMVC*, 2010, doi:10.5244/C.24.12.

[50] S. Johnson and M. Everingham, "Learning effective human pose estimation from inaccurate annotation," in *CVPR*, 2011.

[51] S. Maji, L. Bourdev, and J. Malik, "Action recognition from a distributed representation of pose and appearance," in *CVPR*. IEEE, 2011, pp. 3177–3184.

[52] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001.

[53] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollar, "Microsoft coco: Common objects in context," *arXiv preprint arXiv:1405.0312*, 2014.

[54] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *CVPR*, June 2011, pp. 1385–1392.

[55] M. Kiefel and P. V. Gehler, "Human pose estimation with fields of parts." in *ECCV*, 2014, pp. 331–346.

[56] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *CVPR*. IEEE, 2009, pp. 1014–1021.

[57] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *CVPR*. IEEE, 2014, pp. 1653–1660.

[58] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *CVPR*. IEEE, 2013, pp. 3476–3483.

[59] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-ucsd birds 200," 2010.

[60] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *CVPR*, vol. 2. IEEE, 2006, pp. 1447–1454.

[61] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, vol. 2, 2011.

[62] T. Berg and P. N. Belhumeur, "Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation," in *CVPR*, 2013.

[63] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis, "Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance," in *ICCV*, 2011.

[64] S. Branson, G. Van Horn, S. Belongie, and P. Perona, "Bird species categorization using pose normalized deep convolutional nets," in *BMVC*, 2014.

[65] P. N. Belhumeur, D. W. Jacobs, D. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," in *CVPR*, 2011.

[66] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[67] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based r-cnns for fine-grained category detection," in *ECCV*. Springer, 2014, pp. 834–849.

[68] M. Ren, R. Kiros, and R. Zemel, "Exploring models and data for image question answering," *arXiv preprint arXiv:1505.02074v3*, 2015.

[69] L. Yu, E. Park, A. C. Berg, and T. L. Berg, "Visual madlibs: Fill in the blank image generation and question answering," *arXiv preprint arXiv:1506.00278*, 2015.

[70] M. F. Mateusz Malinowski, Marcus Rohrbach, "Ask your neurons: A neural-based approach to answering questions about images," in *ICCV*, 2015.

[71] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik, "Improving image-sentence embeddings using large weakly annotated photo collections," in *ECCV*. Springer, 2014, pp. 529–545.

[72] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *CVPR*, 29 2009-oct. 2 2009, pp. 1365 –1372.

[73] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *CVPR*. IEEE, 2014.

[74] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[75] B. Iglewicz and D. C. Hoaglin, *How to detect and handle outliers*. Asq Press, 1993, vol. 16.

[76] Y. A. Sheikh, E. A. Khan, and T. Kanade, "Mode-seeking by medoidshifts," in *ICCV*. IEEE, 2007, pp. 1–8.

[77] L. Bourdev, S. Maji, T. Brox, and J. Malik, "Detecting people using mutually consistent poselet activations," in *ECCV*, 2010.

[78] N. Zhang, R. Farrell, F. Iandola, and T. Darrell, "Deformable part descriptors for fine-grained recognition and attribute prediction," in *ICCV*, 2013.

[79] D. Lin, X. Shen, C. Lu, and J. Jia, "Deep lac: Deep localization, alignment and classification for fine-grained recognition," in *CVPR*, 2015, pp. 1666–1674.

[80] Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *ICCV*, 2013.

[81] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars, "Fine-grained categorization by alignments," in *ICCV*, 2013.

[82] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv:1310.1531*, 2013.

[83] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[84] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, pp. 1–42, April 2015.

[85] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "Weakly supervised memory networks," *CoRR*, vol. abs/1503.08895, 2015. [Online]. Available: http://arxiv.org/abs/1503.08895

[86] M.-C. De Marneffe, B. MacCartney, C. D. Manning et al., "Generating typed dependency parses from phrase structure parses," in *Proceedings of LREC*, vol. 6, no. 2006, 2006, pp. 449–454.

[87] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *BMVC*, 2014.

[88] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[89] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[90] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[91] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, "Simple baseline for visual question answering," *arXiv preprint arXiv:1512.02167*, 2015.

[92] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," *CoRR*, vol. abs/1505.04870, 2015. [Online]. Available: http://arxiv.org/abs/1505.04870

[93] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016. [Online]. Available: http://arxiv.org/abs/1602.07332

[94] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[95] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," *arXiv preprint arXiv:1511.02274*, 2015.

[96] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," *arXiv preprint arXiv:1606.00061*, 2016.

[97] K. J. Shih, S. Singh, and D. Hoiem, "Where to look: Focus regions for visual question answering," in *CVPR*, 2016.

[98] T. Tommasi, A. Mallya, B. Plummer, S. Lazebnik, A. C. Berg, and T. L. Berg, "Solving visual madlibs with multiple cues," in *BMVC*, 2016.

[99] I. Ilievski, S. Yan, and J. Feng, "A focused dynamic attention model for visual question answering," *arXiv preprint arXiv:1604.01485*, 2016.

[100] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," in *EMNLP*. ACL, 2016, pp. 457–468.